

1 Syntax

SCSS (.scss) — braces + semicolons, a superset of CSS. **The default.**

```
$c: #3f51b5;
.btn {
  color: $c;
  &:hover { color: #fff; }
}
```

Indented Sass (.sass) — no braces, no semicolons.

```
$c: #3f51b5
.btn
  color: $c
  &:hover
    color: #fff
```

```
/* kept in output */
// stripped from output
```

2 Variables

```
$primary: #3f51b5;
$radius: 4px !default; // overridable
$theme: dark !global; // set outer
```

Scope: top level = global; inside a block = local (shadows, never reassigns). Loop bodies *can* assign outward without !global.

vs. CSS custom properties: \$x is compile-time and erased; --x is runtime and cascades.

```
:root { --primary: #{$primary}; }
// interpolation required!
```

3 Nesting

```
nav {
  ul { margin: 0; } // nav ul
  &:hover { } // nav:hover
  &::before { } // nav::before
  &.active { } // nav.active
  .dark & { } // .dark nav
  &__title { } // nav__title
}
```

& = the **full** parent selector. Nested properties:

```
font: { family: serif; size: 2em; }
```

Keep it shallow — max 3 levels; deep nesting breeds specificity.

4 Partial & Modules

`_name.scss` — leading underscore = partial, never compiled alone.

```
@use "colors"; //
colors.$primary
@use "colors" as c; // c.$primary
@use "colors" as *; // no namespace
@use "theme" with (
  $primary: #e91e63); // configure

@forward "colors" show $primary;
@forward "type" as type-*;
```

Loaded **once**, namespaced, top of file only. `—/_` prefix = private.

@import is deprecated — migrate with sass-migrator module.

5 Mixins

```
@mixin button($bg, $fg: white) {
  background: $bg;
  color: $fg;
}

.a { @include button(#3f51b5); }
.b { @include button(red, $fg: #000); }
```

Variadic + content block:

```
@mixin respond($w) {
  @media (min-width: $w) { @content; }
}

.s { @include respond(768px) {
  width: 280px; } }

@mixin sh($list...) { box-shadow: $list; }
```

6 Functions

```
@use "sass:math";

@function rem($px, $base: 16px) {
  @return math.div($px, $base) * 1rem;
}

.t { font-size: rem(24px); } // 1.5rem
```

Function returns a value — **mixin** emits declarations. Every path must @return.

```
@error "bad arg"; // abort
@warn "deprecated";
@debug $value;
```

7 @extend

```
%msg { padding: 12px; border: 1px solid; }

.ok { @extend %msg; color: green; }
.err { @extend %msg; color: red; }

// → .ok, .err { padding:12px; ... }
```

Merges **selectors**; a mixin copies **declarations**.

%placeholder is the recommended target — it emits nothing on its own.

Caveats: cannot cross @media boundaries; risks selector bloat; lands at the *target's* source position.

Default to mixins.

8 Operators

```
8px + 4px // 12px
8px * 3 // 24px
13 % 5 // 3
math.div(100%, 3) // 33.333%
```

Never use / to divide — deprecated (it is CSS syntax in font: 16px/1.5). Use `math.div()`.

Units: 10px + 5 is an **error**; `math.div(10px, 2px)` → 5 unitless.

Compare: == != < > <= >= **Boolean:** and or not

Truthy: everything except false and null — 0, "", () are **true**.

9 Control Flow

```
@if $t == dark { color: #eee; }
@else if $t == light { color: #222; }
@else { @error "?"; }

@for $i from 1 through 12 { // incl.
  .col-#{$i} { width: $i * 8.33%; }
}
// "to" = exclusive of the end

@each $n, $c in $palette {
  .bg-#{$n} { background: $c; }
}

@while $s < 48px { $s: $s * 1.25; }

if($cond, $a, $b) picks a value, lazily.
```

10 Lists & Maps

```
// lists — 1-INDEXED, immutable
$l: 10px, 20px, 30px;
list.nth($l, 2) // 20px
$l: list.append($l, 40px);

// maps
$bp: (sm: 576px, md: 768px);
map.get($bp, md) // 768px
$bp: map.set($bp, lg, 992px);
map.get($cfg, a, b, c) // nested
```

Immutable — always reassign the result. `map.get` returns null when absent.

11 Built-in Modules

<code>sass:math</code>	<code>math.div()</code> , <code>percentage()</code> , <code>round()</code> , <code>\$pi</code>
<code>sass:color</code>	<code>color.adjust()</code> , <code>scale()</code> , <code>mix()</code>
<code>sass:list</code>	<code>list.nth()</code> , <code>append()</code> , <code>length()</code>
<code>sass:map</code>	<code>map.get()</code> , <code>set()</code> , <code>merge()</code> , <code>keys()</code>
<code>sass:string</code>	<code>string.quote()</code> , <code>unquote()</code> , <code>slice()</code>
<code>sass:meta</code>	<code>meta.type-of()</code> , <code>inspect()</code> , <code>keywords()</code>

Globals deprecated: `darken()` → `color.adjust()`, `map-get()` → `map.get()`, `nth()` → `list.nth()`. Still global & fine: `if()`, `rgb()`, `hsl()`.

12 Compilation

Browsers never run Sass — it **must** compile to plain CSS.

```
# CLI
sass scss/main.scss css/main.css
sass --watch scss:css
sass scss:css --style=compressed \
  --no-source-map
sass --load-path=node_modules ...
```

Dart Sass is the official implementation. **LibSass / node-sass deprecated**, Ruby Sass EOL → `npm i -D sass`.

Vite: built in — just install sass. **Webpack:** `style-loader` ← `css-loader` ← `sass-loader`.

Styles: expanded (default), compressed.