

Sass/SCSS 빠른 참조

변수, 중첩, 믹스인, 함수, 제어 흐름

문법

SCSS vs Sass

```
// SCSS (superset of CSS – uses braces)
.nav { display: flex; }

// Sass (indented – no braces or semicolons)
.nav
  display: flex
```

SCSS가 가장 널리 사용되는 문법

비교

SCSS (.scss)	CSS 호환, 중괄호 및 세미콜론
Sass (.sass)	들여쓰기 기반, 중괄호 없음
출력	둘 다 표준 CSS로 컴파일
권장	SCSS (더 널리 채택, 마이그레이션 쉬움)

변수

정의 및 사용

```
$primary: #3498db;
$spacing: 16px;
$font-stack: "Helvetica", Arial, sans-serif;

.btn {
  color: $primary;
  padding: $spacing;
  font-family: $font-stack;
}
```

변수 범위

```
$color: red; // global
.card {
  $color: blue; // local to .card
  color: $color; // blue
}
.other { color: $color; } // red
```

플래그

!default	아직 정의되지 않은 경우에만 설정
!global	지역 변수를 전역 범위로 승격

중첩

선택자 중첩

```
.nav {
  ul { list-style: none; }
  li { display: inline-block; }
  a { text-decoration: none; }
  &:hover { color: blue; } // & = parent selector
}
```

부모 선택자 (&)

```
.btn {
  &--primary { background: blue; } // BEM: .btn--primary
  &__icon { margin-right: 4px; } // BEM: .btn__icon
  .dark & { color: white; } // .dark .btn
}
```

속성 중첩

```
.box {
  border: { width: 1px; style: solid; color: #ccc; }
  // compiles to: border-width, border-style, border-color
}
```

믹스인

정의 및 포함

```
@mixin flex-center($direction: row) {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: $direction;
}

.hero { @include flex-center(column); }
```

콘텐츠 블록

```
@mixin responsive($breakpoint) {
  @media (min-width: $breakpoint) { @content; }
}

.sidebar {
  width: 100%;
  @include responsive(768px) { width: 300px; }
}
```

믹스인 기능

@mixin name(\$args)	재사용 가능한 스타일 블록 정의
@include name()	믹스인 사용
기본 파라미터	선택적 파라미터를 위한 \$arg: value
\$args...	가변 인수 (나머지 파라미터)
@content	호출자의 콘텐츠 블록 삽입

함수

사용자 정의 함수

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

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

내장 함수

darken(\$color, 10%)	색상 어둡게
lighten(\$color, 10%)	색상 밝게
mix(\$c1, \$c2, 50%)	두 색상 혼합
rgba(\$color, 0.5)	알파 채널 설정
math.div(\$a, \$b)	나눗셈 (/ 대체)
math.round(\$n)	숫자 반올림
string.quote(\$s)	문자열에 따옴표 추가
if(\$cond, \$t, \$f)	인라인 조건

확장

Extend 및 플레이스홀더

```
%flex-center { // placeholder – not emitted
  display: flex;
  justify-content: center;
  align-items: center;
}

.hero { @extend %flex-center; }
.modal { @extend %flex-center; }
// Both share one CSS rule via selector grouping
```

Extend vs 믹스인

@extend	선택자 그룹화 – 더 작은 CSS 출력
@mixin	선언 복사 – 인수 지원
% 플레이스홀더	확장 전용 (미사용 시 출력 안 됨)
권장	파라미터화된 스타일엔 믹스인 선호

부분 파일 및 가져오기

파일 구성

```
// _variables.scss (partial – not compiled alone)
$primary: #3498db;

// main.scss
@use "variables"; // modern: namespaced
.btn { color: variables.$primary; }

@use "variables" as v; // alias
.btn { color: v.$primary; }
```

모듈 시스템

@use 'file'	네임스페이스와 함께 모듈 로드
@use 'file' as *	네임스페이스 없이 로드
@use 'file' as alias	사용자 정의 네임스페이스
@forward 'file'	모듈 멤버 재내보내기
_partial.scss	독립적으로 컴파일되지 않는 파일

@import는 더 이상 사용되지 않음 – @use와 @forward 사용

제어 흐름

조건문

```
@mixin theme($mode) {
  @if $mode == dark {
    background: #333; color: #fff;
  } @else {
    background: #fff; color: #333;
  }
}
```

반복문

```
@for $i from 1 through 4 {
  .col-#{ $i } { width: 25% * $i; }
}

@each $name, $color in (primary: blue, danger: red) {
  .text-#{ $name } { color: $color; }
}
```

지시어

@if / @else if / @else	조건 로직
@for \$i from a through b	숫자 반복 (포함)
@for \$i from a to b	숫자 반복 (끝 미포함)
@each \$item in \$list	리스트 또는 맵 반복
@while	조건이 참인 동안 반복
#{\$var}	선택자/속성에서 보간

맵 및 리스트

맵

```
$colors: (primary: #3498db, danger: #e74c3c, success: #2ecc71);

.alert { color: map.get($colors, danger); }

@each $name, $color in $colors {
  .bg-#{ $name } { background: $color; }
}
```

리스트

```
$sizes: 8px 16px 24px 32px;
.box { padding: list.nth($sizes, 2); } // 16px
```

Sass/SCSS 빠른 참조

맵 및 리스트 함수

<code>map.get(\$map, \$key)</code>	키로 값 가져오기
<code>map.merge(\$m1, \$m2)</code>	두 맵 병합
<code>map.keys(\$map)</code>	모든 키 목록
<code>map.has-key(\$map, \$key)</code>	키 존재 여부 확인
<code>list.nth(\$list, \$n)</code>	인덱스의 항목 가져오기 (1부터)
<code>list.length(\$list)</code>	항목 수
<code>list.append(\$list, \$val)</code>	리스트에 항목 추가

일반 패턴

반응형 브레이크포인트

```
$breakpoints: (sm: 576px, md: 768px, lg: 992px, xl: 1200px);

@mixin bp($name) {
  @media (min-width: map.get($breakpoints, $name)) {
    @content;
  }
}

.sidebar { width: 100%; @include bp(md) { width: 300px; } }
```

유틸리티 생성기

```
$spaces: (0: 0, 1: 4px, 2: 8px, 3: 16px, 4: 32px);
@each $key, $val in $spaces {
  .mt-#{$key} { margin-top: $val; }
  .mb-#{$key} { margin-bottom: $val; }
  .p-#{$key} { padding: $val; }
}
```

다크 모드

```
@mixin dark { @media (prefers-color-scheme: dark) { @content; } }
body {
  background: #fff;
  @include dark { background: #1a1a1a; color: #eee; }
}
```