

# SASS/SCSS 快速参考

变量、嵌套、Mixin、函数与控制流

## 语法

### SCSS 与 Sass

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

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

SCSS 是最常用的语法

## 对比

**SCSS (.scss)** 兼容 CSS，使用大括号和分号  
**Sass (.sass)** 缩进式，无大括号  
**输出** 两者均编译为标准 CSS  
**推荐** SCSS (采用更广泛，迁移更容易)

## 变量

### 定义与使用

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

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

### 变量作用域

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

## 标志

**!default** 仅在未定义时赋值  
**!global** 将局部变量提升为全局作用域

## 嵌套

### 选择器嵌套

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

### 父选择器 (&)

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

## 属性嵌套

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

## Mixin

### 定义与引入

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

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

## 内容块

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

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

## Mixin 特性

**@mixin name(\$args)** 定义可复用样式块  
**@include name()** 使用 Mixin  
**Default params** 'Sarg: value' 用于可选参数  
**\$args...** 可变参数 (rest params)  
**@content** 注入调用方的内容块

## 函数

### 自定义函数

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

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

### 内置函数

**darken(\$color, 10%)** 加深颜色  
**lighten(\$color, 10%)** 减淡颜色  
**mix(\$c1, \$c2, 50%)** 混合两种颜色  
**rgba(\$color, 0.5)** 设置透明度  
**math.div(\$a, \$b)** 除法 (替代 '/')  
**math.round(\$n)** 四舍五入  
**string.quote(\$s)** 为字符串添加引号  
**if(\$cond, \$t, \$f)** 内联条件

## 继承

### Extend 与占位符

```
%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 Mixin

**@extend** 合并选择器，CSS 输出更小  
**@mixin** 复制声明，支持参数  
**% placeholder** 仅用于 extend (未使用时不输出)  
**推荐** 参数化样式优先使用 Mixin

## 局部文件与导入

### 文件组织

```
// 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; }
```

## 模块系统

**@use 'file'** 加载带命名空间的模块  
**@use 'file' as \*** 加载但不使用命名空间  
**@use 'file' as alias** 自定义命名空间  
**@forward 'file'** 重新导出模块成员  
**\_partial.scss** 不单独编译的局部文件  
**@import** 已弃用，请改用 @use 和 @forward

## 控制流

### 条件语句

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

### 循环

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

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

## 指令

**@if / @else if / @else** 条件逻辑  
**@for \$i from a through b** 数值循环 (含两端)  
**@for \$i from a to b** 数值循环 (不含末端)  
**@each \$item in \$list** 遍历列表或 Map  
**@while** 条件为真时循环  
**#{ \$var }** 在选择器 / 属性中插值

## Map 与列表

### Map

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

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

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

### 列表

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

## Map 与列表函数

**map.get(\$map, \$key)** 按键获取值  
**map.merge(\$m1, \$m2)** 合并两个 Map  
**map.keys(\$map)** 所有键的列表  
**map.has-key(\$map, \$key)** 检查键是否存在  
**list.nth(\$list, \$n)** 获取第 n 个元素 (从 1 开始)  
**list.length(\$list)** 元素数量  
**list.append(\$list, \$val)** 向列表追加元素

## 常用模式

### 响应式断点

```
$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; } }
```

## 工具类生成器

```
$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; }
}
```

## 深色模式

```
@mixin dark { @media (prefers-color-scheme: dark) { @content; } }

body {
  background: #fff;
  @include dark { background: #1a1a1a; color: #eee; }
}
```