

Sass/SCSS Riferimento Rapido

Variabili, annidamento, mixin, funzioni, flusso di controllo

Sintassi

SCSS vs Sass

```
// SCSS (superset di CSS – usa le parentesi graffe)
.nav { display: flex; }

// Sass (indentato – senza parentesi o punto e virgola)
.nav
  display: flex
```

SCSS è la sintassi più utilizzata

Confronto

SCSS (.scss)	Compatibile CSS, parentesi e punto e virgola
Sass (.sass)	Basato sull'indentazione, senza parentesi
Output	Entrambi compilano in CSS standard
Raccomandato	SCSS (adozione più ampia, migrazione più facile)

Variabili

Definizione e Utilizzo

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

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

Scope delle Variabili

```
$color: red; // globale
.card {
  $color: blue; // locale a .card
  color: $color; // blue
}
.other { color: $color; } // red
```

Flag

!default	Imposta solo se non ancora definita
!global	Promuove variabile locale a scope globale

Annidamento

Annidamento dei Selettori

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

Selettore Genitore (&)

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

Annidamento delle Proprietà

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

Mixin

Definisci e Includi

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

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

Blocchi di Contenuto

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

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

Funzionalità dei Mixin

@mixin name(\$args)	Definisce blocco di stile riutilizzabile
@include name()	Usa il mixin
Parametri con default	\$arg: value per parametri opzionali
\$args...	Argomenti variabili (rest params)
@content	Inietta il blocco di contenuto del chiamante

Funzioni

Funzioni Personalizzate

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

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

Funzioni Integrate

darken(\$color, 10%)	Scurisce un colore
lighten(\$color, 10%)	Schiarisce un colore
mix(\$c1, \$c2, 50%)	Mescola due colori
rgba(\$color, 0.5)	Imposta il canale alpha
math.div(\$a, \$b)	Divisione (sostituisce /)
math.round(\$n)	Arrotonda un numero
string.quote(\$s)	Aggiunge virgolette alla stringa
if(\$cond, \$t, \$f)	Condizionale inline

Extends

Extend e Placeholder

```
%flex-center { // placeholder – non emesso
  display: flex;
  justify-content: center;
  align-items: center;
}

.hero { @extend %flex-center; }
.modal { @extend %flex-center; }
// Entrambi condividono una regola CSS via raggruppamento
```

Extend vs Mixin

@extend	Raggruppa selettori — output CSS più piccolo
@mixin	Copia dichiarazioni — supporta argomenti
% placeholder	Solo per extend (non emesso se non usato)
Raccomandazione	Preferire mixin per stili parametrizzati

Parziali e Import

Organizzazione dei File

```
// _variables.scss (parziale – non compilato da solo)
$primary: #3498db;

// main.scss
@use "variables"; // moderno: con namespace
.btn { color: variables.$primary; }

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

Sistema di Moduli

@use 'file'	Carica modulo con namespace
@use 'file' as *	Carica senza namespace
@use 'file' as alias	Namespace personalizzato
@forward 'file'	Ri-esporta i membri del modulo
_partial.scss	File non compilato indipendentemente

@import è deprecato — usa @use e @forward

Flusso di Controllo

Condizionali

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

Cicli

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

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

Direttive

@if / @else if / @else	Logica condizionale
@for \$i from a through b	Ciclo numerico (incluso)
@for \$i from a to b	Ciclo numerico (fine esclusa)
@each \$item in \$list	Itera lista o mappa
@while	Ciclo mentre la condizione è vera
#{\$var}	Interpolazione in selettori/proprietà

Mappe e Liste

Mappe

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

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

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

Liste

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

Sass/SCSS Riferimento Rapido

Funzioni su Mappe e Liste

<code>map.get(\$map, \$key)</code>	Ottieni valore per chiave
<code>map.merge(\$m1, \$m2)</code>	Unisce due mappe
<code>map.keys(\$map)</code>	Lista di tutte le chiavi
<code>map.has-key(\$map, \$key)</code>	Verifica se la chiave esiste
<code>list.nth(\$list, \$n)</code>	Ottieni elemento all'indice (base 1)
<code>list.length(\$list)</code>	Numero di elementi
<code>list.append(\$list, \$val)</code>	Aggiunge elemento alla lista

Pattern Comuni

Breakpoint Responsive

```
$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; } }
```

Generatore di Utility

```
$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; }
}
```

Dark Mode

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