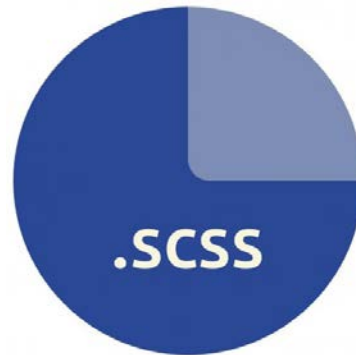


# Sass - SCSS



# CSS dal 1996

---

- “Cascading Style Sheet (CSS) is the language for describing the presentation of Web pages, including colors, layout, and fonts.
- CSS can be used with any XML-based markup language.
- Separation of structure (or content) from presentation.
- The separation of HTML from CSS makes it easier to maintain sites and share style sheets across pages.”

[W3C]

# Alcune limitazioni

---

- Tante limitazioni!
- Tra le più comuni:
  1. Il committente cambia idea a CSS completato
  2. File pieni di commenti
  3. Molte chiamate HTTP per caricare lo stile della pagine Web

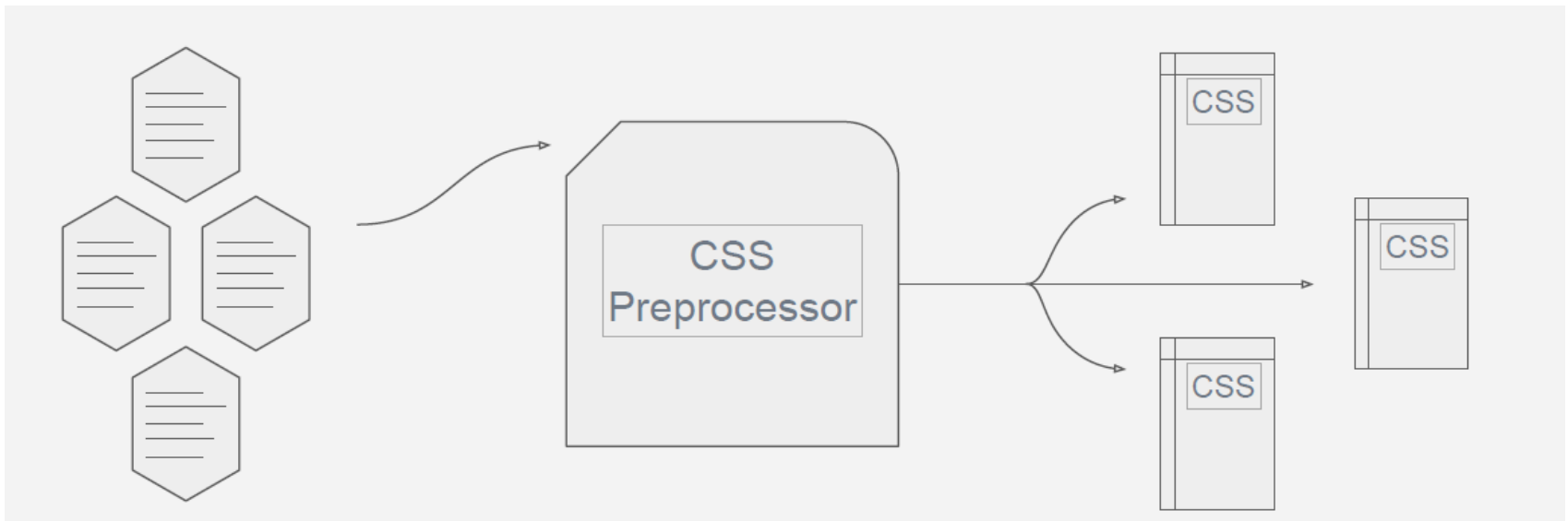
# La soluzione: Preprocessor & Transpiler

---

- Vengono utilizzati per:
  - Aggiungere feature non disponibili nel linguaggio di partenza
  - Mantenere il codice DRY e manutenibile.
  - Eseguire controlli a tempo di “compilazione”
- Alcuni Preprocessor/Transpiler:
  - Per **HTML**: HAML, Pug, Slim, ecc
  - Per **CSS**: SASS, Less, Stylus, PostCSS, ecc
  - Per **Javascript**: **TypeScript**, DART, ecc

# CSS Preprocessor

- Un preprocessore CSS serve a definire fogli di stile con una forma più semplice, completa e potente rispetto ai CSS e a generare file CSS ottimizzati, aggregando le strutture definite anche in modo complesso



# SASS: definizione



- SASS sta per ***Syntactically Awesome Style Sheets***
1. Sass is an extension of CSS that adds power and elegance to the basic language
  2. It allows you to use variables, nested rules, mixins, inline imports, and more
  3. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly

[<https://sass-lang.com/>]

# SASS: installazione



- Installazione Anywhere (Standalone)
  - E' possibile installare Sass su Windows, Mac e Linux
  - Download del pacchetto specifico per il Sistema operativo da GitHub, aggiungere il percorso alla variabile d'ambiente **PATH**
  - Non ci sono dipendenze esterne o altro che debba essere installato

# SASS: installazione



- Installazione Anywhere (npm)
  - E' possibile Usare Node.js Package Manager, eseguendo  
**npm install -g sass**
  - Potrebbe essere più lento delle altre possibili soluzioni



# SASS: installazione



- Installazione su Windows (Chocolatey)
  - Se si usa Chocolatey package manager per Windows, è possibile installare DART Sass eseguendo  
**choco install sass**
- Install on Mac OS X (Homebrew)
  - Se si usa Homebrew package manager per Mac OS X, è possibile installare DART Sass eseguendo  
**brew install sass/sass/sass**

# SASS: caratteristiche tecniche

---



- Estensioni del linguaggio (variables, nesting, import, mixins, eccetera)
- Direttive di controllo (if, for, each, while)
- Funzioni ausiliarie (colori, numeri, liste, eccetera)
- Output formattato e personalizzabile

# SASS: sintassi



- La sintassi di Sass si discosta dalle regole base del CSS
- Questo rende la vita difficile agli sviluppatori che devono ri-convertire il loro modo di lavorare in CSS (e i loro progetti) verso questo nuovo linguaggio

# SCSS

---

- **SCSS** sta per *Sassy* - CSS
- Fully CSS-compatible

# CSS

---

```
#sidebar {  
    width: 30%;  
    background-color: #ffaana;  
}
```

# SASS vs SCSS

```
#sidebar  
width: 30%  
background-color: #faa
```

SASS



```
#sidebar {  
width: 30%;  
background-color: #faa;  
}
```

CSS

```
#sidebar {  
width: 30%;  
background-color: #faa;  
}
```

SCSS



# SCSS: Command line

---

- Run SASS

```
sass input.scss output.css
```

- Watch file

```
sass --watch input.scss:output.css
```

- Watch directory

```
sass --watch app/sass:public/stylesheets
```

# SCSS: Command line - Output

---

- Sass consente di definire lo stile dell'output "CSS" che desideriamo ottenere

```
sass --watch style.scss:style.css --style compressed
```

- Usando il flag `--style` da linea di comando con le seguenti opzioni:
  1. `nested`
  2. `expanded`
  3. `compact`
  4. `compressed`



# SCSS: Command line - Output

## Nested

```
#main {  
  color: #fff;  
  background: #000; }  
#main p {  
  width: 10em; }
```

## Expanded

```
#main {  
  color: #fff;  
  background: #000; }  
#main p {  
  width: 10em; }
```

## Compact

```
#main { color: #fff; background: #000; }  
#main p { width: 10em; }
```

## Compressed

```
#main{color:#fff;background:#000;}#main p{width:10em;}
```

# Nested Rules

```
#main p {  
  color: #0f0;  
  width: 97%;  
  
  .redbox {  
    background-color: #f00;  
    color: #000;  
  }  
}
```

SCSS



```
#main p {  
  color: #0f0;  
  width: 97%;  
}  
  
#main p .redbox {  
  background-color: #f00;  
  color: #000;  
}
```

CSS

# Parent Selector «&»

```
a {  
  font-weight: bold;  
  text-decoration: none;  
  &:hover { color: #FFF;  
}  
body.firefox & {  
  font-weight: normal;  
}  
}
```

SCSS



CSS

```
a {  
  font-weight: bold;  
  text-decoration: none;  
}  
a:hover {  
  color: #FFF;  
}  
body.firefox a {  
  font-weight: normal;  
}
```

# Parent Selector «&»

```
#main {  
  color: black;  
  a {  
    font-weight: bold;  
    &:hover { color: red; }  
  }  
}
```

SCSS



```
#main {  
  color: black;  
}  
#main a {  
  font-weight: bold;  
}  
#main a:hover {  
  color: red;  
}
```

CSS

# Parent Selector «&»

```
#main {  
  color: black;  
  &-sidebar {  
    border: 1px solid; }  
}
```

SCSS




```
#main {  
  color: black;  
}  
#main-sidebar {  
  border: 1px solid;  
}
```


CSS

# Variabili

```
#main {  
  $width: 5em !global;  
  width: $width;  
}  
  
#sidebar {  
  width: $width; }  
}
```



```
#main {  
  width: 5em;  
}  
#sidebar {  
  width: 5em;  
}
```



**!global** indica una variabile che può essere usata globalmente. Una nuova assegnazione ne modifica il valore da quel punto in poi

# Variabili

```
#main {  
  $width: 5em !global;  
  width: $width;  
}  
  
#sidebar {  
  width: $width;  
}  
}
```

SCSS



```
#main {  
  width: 5em;  
}  
  
#sidebar {  
  width: 5em;  
}
```

CSS

**!global** indica una variabile che può essere usata globalmente. Una nuova assegnazione ne modifica il valore da quel punto in poi

# Variabili: Tipi supportati

---

I tipi supportati nativamente da SASS sono: numeri, stringhe, colori, booleani, null, liste e mappe chiave-valore

```
$numero: 5;  
$stringa: "Stringa di prova";  
$colore: #ff00ff;  
$boolean: true;  
$valoreNulla: null;  
$lista: ("elem1", "elem2", "elem3");  
$mappa: (k1: v1, k2: v2, k3: v3);
```



# Operazioni

Operazioni supportate: + - \* /

[**Nota.** L'espressione viene valutata solo se è presente sintassi SCSS]

```
p {  
  font: 10px/8px;           /* Plain CSS, no division */  
  $width: 1000px;  
  width: $width/2;          // Uses a variable, does division  
  width: round(1.5)/2;      // Uses a function, does division  
  height: (500px/2);        // Uses parentheses, does division  
  margin-left: 5px + 8px/2px; // Uses +, does division  
  font: (italic bold 10px/8px); // Parentheses founded. Does division  
}
```

# Operazioni

```
p {  
  $marg: 10px -5px;  
  margin: $marg;  
}
```

SCSS

```
p {  
  margin: 10px -5px;  
}
```

CSS

```
p {  
  $marg: 10px-5px;  
  margin: $marg;  
}
```

SCSS

```
p {  
  margin: 5px;  
}
```

CSS

# Interpolazione #{} ---

```
p {  
  $font-size: 12px;  
  $line-height: 30px;  
  font: #{ $font-size } / #{ $line-height };  
}
```

SCSS

Se voglio mantenere la sintassi CSS utilizzando variabili SCSS, occorre utilizzare un carattere di interpolazione #{}.

```
p {  
  font: 12px/30px;  
}
```

CSS

Senza l'interpolazione? Senza l'interpolazione delle variabili SCSS sarebbe stata eseguita la divisione.

# Operazioni

```
$myvar: superclass;
```

SCSS

```
p.#{$myvar} {  
  a##{mynewid} {  
    color: #0F0;  
    &:hover { background:#000; }  
  }  
  &:after {  
    content: hello + -world;  
  }  
}
```

```
p.superclass a#mynewid {  
  color: #0F0;  
}  
p.superclass a#mynewid:hover {  
  background: #000;  
}  
p.superclass:after {  
  content: hello-world;  
}
```

CSS

# DRY

---

- “Don’t Repeat Yourself”
- SCSS offre un insieme di direttive che consentono di rendere il codice più **ordinato**, **modulare** e **scalabile** rispetto al CSS

# Direttive: @import

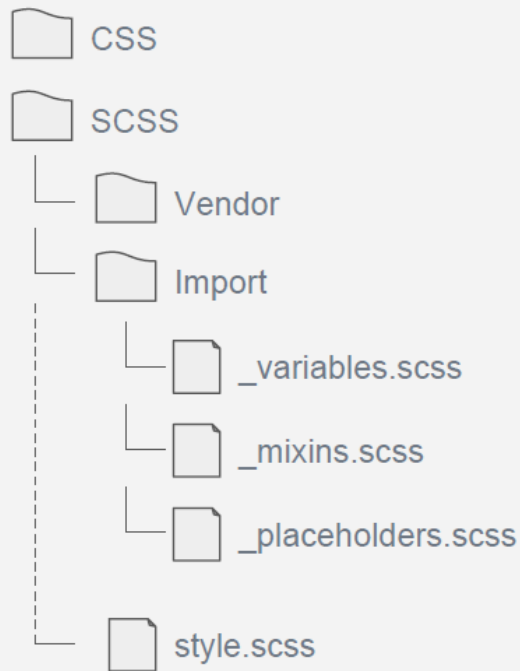
---

Consente di importare file SCSS esterni col fine di utilizzare regole definite al loro interno, rendendo tutto estremamente modulare

```
@import "foo.scss";  
@import "foo" screen;  
@import "http://foo.com/bar";  
@import url(foo);
```

[**Nota.** Se non vuole essere creato un file css corrispondente, aggiungere underscore “\_” al nome del file: `_variables.scss`]

# Struttura cartelle e file: un esempio



# Directive: @media

```
$media: screen;
$feature: -webkit-min-device-pixel-ratio;
$value: 1.5;

@media #{ $media } and ( $feature: $value ) {
  .sidebar {
    width: 500px;
  }
}
```

SCSS

```
@media screen and (-webkit-min-device-pixel-ratio: 1.5) {
  .sidebar {
    width: 500px;
  }
}
```

CSS



# Directive: @extend

```
<div class="error seriousError">  
  Oh no! You've been hacked!  
</div>
```

HTML

```
.error {  
  border: 1px #f00;  
  background-color: #fdd;  
}  
  
.seriousError {  
  @extend .error;  
  border-width: 3px;  
}
```

SCSS

```
.error, .seriousError {  
  border: 1px #f00;  
  background-color: #fdd;  
}  
  
.seriousError {  
  border-width: 3px;  
}
```

CSS

# Directive: @extend

```
.error {  
  border: 1px #f00;  
  background-color: #fdd;  
}  
.error.intrusion {  
  background-image: url("hacked.png");  
}  
.seriousError {  
  @extend .error;  
  border-width: 3px;  
}
```

SCSS

```
.error, .seriousError {  
  border: 1px #f00;  
  background-color: #fdd;  
}  
  
.error.intrusion, .intrusion.seriousError {  
  background-image: url("hacked.png");  
}  
  
.seriousError {  
  border-width: 3px;  
}
```

CSS

# Directive: @extend

## Placeholder selector “%”

```
%bordered {  
  border: 1px solid #CCC;  
}  
.padded {  
  padding: 10px;  
}  
.fancy-box {  
  @extend %bordered;  
  @extend .padded;  
}
```

SCSS

```
.fancy-box {  
  border: 1px solid #CCC;  
}  
  
.padded, .fancy-box {  
  padding: 10px;  
}
```

CSS

# Direttive di controllo

---

- Le direttive di controllo (control directives), aggiungono un ulteriore livello di dinamicità a SCSS, consentendo di includere regole solo al verificarsi di determinate condizioni
  - `@if`
  - `@for`
  - `@each`
  - `@while`

# @if

```
$type: monster;
p {
  @if $type == ocean {
    color: blue;
  } @else if $type == monster {
    color: green;
  } @else {
    color: black;
  }
}
```

SCSS

```
p {
  color: green;
}
```

CSS

# @for

```
@for $i from 1 through 3 {  
  .item-#{ $i } {  
    width: 2em * $i;  
  }  
}
```

SCSS

```
.item-1 {  
  width: 2em;  
}  
  
.item-2 {  
  width: 4em;  
}  
  
.item-3 {  
  width: 6em;  
}
```

CSS

# @each

```
@each $animal in puma, lion, salamander {  
  .#{$animal}-icon {  
    background-image: url('/#{$animal}.png');  
  }  
}
```

SCSS

```
.puma-icon {  
  background-image: url("/puma.png");  
}  
  
.lion-icon {  
  background-image: url("/lion.png");  
}  
  
.salamander-icon {  
  background-image: url("/salamander.png");  
}
```

CSS

# @while

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

SCSS

```
.item-6 {  
  width: 12em;  
}  
  
.item-4 {  
  width: 8em;  
}  
  
.item-2 {  
  width: 4em;  
}
```

CSS



# Riuso del codice

---

- Quali direttive principali abbiamo visto per il riuso del codice?
  1. **@import** - importa un file SCSS
  2. **@extend** - estende una porzione di codice SCSS precedentemente definita

# Mixin

---

- Consente di definire “stili” (regole) che possono essere riutilizzati
- Accetta argomenti in input, col fine di definire procedure che aggiungano un buon livello di variabilità al risultato ottenuto

# Mixin - @mixin/@include

```
@mixin large-text {  
  font: {  
    family: Arial;  
    size: 20px;  
    weight: bold;  
  }  
  color: #ff0000;  
}
```

SCSS

```
.page-title {  
  @include large-text;  
  padding: 4px;  
  margin-top: 10px;  
}
```

```
.page-title {  
  font-family: Arial;  
  font-size: 20px;  
  font-weight: bold;  
  color: #ff0000;  
  padding: 4px;  
  margin-top: 10px;  
}
```

CSS

# Mixin - @mixin/@include

```
@mixin sexy-border($color, $width) {  
  border: {  
    color: $color;  
    width: $width;  
    style: dashed;  
  }  
}  
  
p { @include sexy-border(blue, 2px); }
```

SCSS

```
p {  
  border-color: blue;  
  border-width: 2px;  
  border-style: dashed;  
}
```

CSS

- La presenza di argomenti in @mixin (non è un semplice @extend), cambia completamente la visione del riuso del codice in un linguaggio come il CSS, in cui il riuso è assente

# Mixin – input arguments

- Definire valori di default

```
@mixin sexy-border($color, $width: 2px) {
```

- Assegnare valori a certi argomenti in chiamata (per nome argomento)

```
p { @include sexy-border($color: blue); }  
h1 { @include sexy-border($color: blue, $width: 2px); }
```

# Mixin – input arguments

- Definire un numero di argomenti sconosciuto

```
@mixin box-shadow($shadows...) {  
  -moz-box-shadow: $shadows;  
  -webkit-box-shadow: $shadows;  
  box-shadow: $shadows;  
}  
  
.shadows {  
  @include box-shadow(0px 4px 5px #666, 2px 6px 10px #999);  
}
```

# Mixin – input arguments

- Passare un numero di argomenti sconosciuto (in chiamata)

```
@mixin colors($text, $background: #F00, $border: #F00) {  
  color: $text;  
  background-color: $background;  
  border-color: $border;  
}  
  
$values: #ff0000, #00ff00, #0000ff;  
.primary {  
  @include colors($values...);  
}  
  
$value-map: (text: #0f0, background: #00f);  
.secondary {  
  @include colors($value-map...);  
}
```

# SASS Function

---

- Esistono un insieme di funzioni messe a disposizione da Sass per la manipolazione dei dati, esempio:
  - colori
  - stringhe (quote, str-length, to-upper-case, ...)
  - numeri (max, min, random, round, ...)
  - ...



# SASS Function - @function/@return

```
@function pow($number, $exponent) {  
  $value: 1;  
  
  @if $exponent > 0 {  
    @for $i from 1 through $exponent {  
      $value: $value * $number;  
    }  
  }  
  
  @return $value; //+px  
}  
  
.foo {  
  width: pow(20, 2) * 1px; // 400px  
}
```

SCSS

```
.foo {  
  width: 400px;  
}
```

CSS

# Riferimenti

---

- Sass documentation  
(<https://sass-lang.com/documentation>)
- Sass Functions  
(<https://sass-lang.com/documentation/at-rules/function>)
- Sass Built-In Modules  
(<https://sass-lang.com/documentation/modules>)
- CSS Tricks - snippets  
(<https://css-tricks.com/snippets/sass/>)
- CSS Tricks - SCSS style guide  
(<https://css-tricks.com/sass-style-guide/>)

# Riferimenti

---

- The Sass Way  
(<http://thesassway.com/>)
- Sitepoint  
(<https://www.sitepoint.com/8-tips-help-get-best-sass/>)
- Nettuts  
(<https://code.tutsplus.com/tutorials/search/sass>)
- Sass Meister – editor  
(<https://www.sassmeister.com/>)