



SASS

CHE COS'È UN PREPROCESSORE CSS?

- Potenzia gli strumenti per creare fogli di stile
- Forma dichiarativa più completa e potente del css
- Genera il file css, pre-processando e aggregando la struttura del file

SASS

- Sintassi più semplificata rispetto al CSS
- Il codice Sass viene scritto dallo sviluppatore, ed in seguito trasformato da un software esterno in codice CSS
- Supporto delle variabili
- Indentazione
- Partials & Import
- Mixins
- Ereditarietà
- Operatori matematici

NIDIFICAZIONE

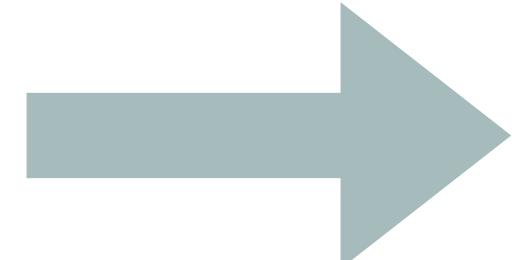
Sass ti permette di nidificare il tuo codice in modo da limitare lo scope della tua regola in un determinato spazio.

```
.nav{  
  ul{  
    float: left;  
  }  
}
```

Assegna la regola solamente agli elementi contenuti dentro ad elementi <nav>

ESEMPIO

```
#main {  
    background-color: red;  
}  
  
#main div {  
    color: blue  
}  
  
#main div .red {  
    color: red;  
}  
  
#main div .red a {  
    text-decoration: none;  
}
```

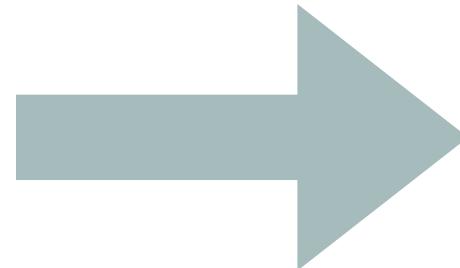


```
#main {  
    background-color: red;  
}  
  
div {  
    color: blue  
}  
  
.red {  
    color: red;  
}  
  
a {  
    text-decoration: none;  
}
```

KEYWORD &

- Ci sono situazioni in cui la sintassi precedente non è sufficiente
- La & in fase di compilazione viene sostituita con il selettore padre

```
#a {  
    color: red;  
    font-size: 1.5em;  
    text-decoration: none;  
}  
  
#a:hover {  
    text-decoration: underline;  
}
```



```
#a {  
    color: red;  
    font-size: 1.5em;  
    text-decoration: none;  
}&:hover {  
    text-decoration: underline;  
}
```

KEYWORD &

```
#main {  
  color: black;  
  &-sidebar {  
    border: 1px solid;  
  }  
  body.firefox & {  
    line-height: 1.5em  
  }  
}
```



```
#main {  
  color: black;  
}  
#main-sidebar {  
  border: 1px solid;  
}  
body.firefox #main{  
  line-height: 1.5em  
}
```

EREDITARIETÀ

Puoi assegnare o ereditare tutte le regole che sono applicate ad un certo elemento html ad un altro elemento, modificando o aggiungendo alcune regole in più

La direttiva `@extend` permette di indicare al compilatore che una classe eredita tutte le proprietà di un'altra precedentemente definita.

```
.tabella{  
    border: 1px solid #fff;  
    border-radius: 5px;  
    background: #333;  
}  
.menu{  
    @extends .tabella;  
    color: #000;  
}  
.eventi{  
    @extends .tabella;  
    color: #34e34a;  
}
```

MIXINS

Puoi dichiarare un set di regole che vuoi applicare a più elementi html, e quando hai bisogno di usarlo basta importarlo con il comando `@include`.

I Mixins sono vere e proprie funzioni, infatti possono anche accettare parametri di input, che poi puoi usare per creare regole simili ma con valori diversi

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius;  
}  
.box { @include border-radius(10px); }
```

MIXINS

```
@mixin baseButton($bgColor)
{
    -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;
    padding: 10px 20px;
    display: inline-block;
    text-transform: uppercase;
    background: $bgColor;
    outline: 1px solid $bgColor;
    box-shadow: 0 0 10px $bgColor;
    color: white;
    border: 1px solid white;
    &:hover {
        color: $bgColor;
        background: white;
        box-shadow: none;
    }
}
button.green{
    @include baseButton(green);
}
button.red{
    @include baseButton(red);
}
```

SUPPORTO DELLE VARIABILI

Sass ti permette di usufruire di variabili per salvare le tue regole CSS, niente più ripetizioni di dichiarazioni di fonts, colori, grandezze, etc.

Sass puoi dichiarare tutte le tue variabili all'inizio del tuo file e queste saranno accessibili sempre nel tuo codice

```
$colore-link: #0044fe;  
nav a {  
    color: $colore-link;  
}
```

TIPI VARIABILI

Tipo di variabile	Descrizione
numero	es: 1, 2, 3 ma anche 10px o 2em
stringa di testo	es: ‘hello’, “goodbye” ma anche senza apici quando non collide con altri tipi di variabili
colore	es: red, blue, oppure #FF00A2 o rgba(10,0,230,0.5), da notare l'assenza degli apici
boolean	true o false
valore nullo	null
lista di valori	separati dallo spazio o dalla virgola (es: 10px 2px 3px oppure Helvetica, Arial, sans-serif)
mappa chiave valore	es: (nome: Sandro, cognome: Paganotti)

OPERATORI MATEMATICI

Ti permette di eseguire operazioni matematiche, usate nei più comuni linguaggi di programmazione, come: +, -, *, /, etc

Può risultare utile soprattutto quando stai scrivendo regole per le dimensioni dei tuoi elementi html.

```
$scala: 1.5;  
$larghezza: 1000px;  
.svg{  
  width: $larghezza / $scala;  
}
```

Per esempio se vuoi avere tutti i tuoi elementi html con delle dimensioni in scala, potresti dichiarare una variabile “scale” e poi dividere i tuoi elementi per la scala

PARTIALS E IMPORTS

Permette di creare porzioni di codice che possono essere riutilizzati, portandoti alla creazione di un codice modulare.

Per creare un partial (modulo), basta salvare un determinato set di regole in un file Sass, questo potrà essere importato in qualsiasi file tramite il comando **@import**.

PER CONVERTIRE SASS A CSS

Per usare SASS prima dobbiamo configurare la nostra macchina.

1. Dobbiamo installare prima Ruby
2. Installare SASS
3. Attraverso riga di comando lanciare la conversione del file

Oggi per semplificare le prove utilizziamo:

<https://www.sassmeister.com/>

ESERCIZIO

- Definisci tre variabili: colore principale, colore secondario e colore testo
- Creare tre colonne: quella a destra con colore di sfondo secondario
- Ogni colonna contiene un testo che dovrà essere colore testo

CONCLUSIONE

linkedin: /giuliamaddalena - giuliamadda09@gmail.com