Corso SASS

Syntactically Awesome Style Sheets

per tutti

Scopo del corso

SASS è uno strumento molto usato in ambito Web Design. Conoscerlo ed esserne padroni permette di ottimizzare i processi di creazione di CSS e pagine web.

All'interno di questo corso proverò a guidarvi nell'uso di queste tecnologia largamente diffusa ed utilizzata.

SASS - di cosa si tratta?

SASS, acronimo di Syntactically Awesome Style Sheets, è un linguaggio di script per la formattazione a cascata dei fogli di stile che offre funzionalità avanzate ed è molto utilizzato in sviluppo web.

SASS - perché usarlo?

Ci permette di scrivere fogli di stile CSS in modo più semplice e veloce, con meno codice e più funzionalità.

Ad esempio, SASS ci permette di:

- Usare variabili
- Usare funzioni
- Usare cicli
- Usare condizioni
- Usare import

SASS - come installarlo

SASS è un preprocessore, quindi per usarlo dobbiamo installarlo sul nostro computer.

Per farlo, dobbiamo installare NodeJS e poi installare SASS tramite il comando:

```
npm install -g sass
```

Oppure tramite chocolatey

```
choco install sass
```

SASS - come usarlo

Una volta installato, possiamo usare SASS in due modi:

- Usando il comando `sass` da terminale
- Usando un IDE che supporta SASS

Uso da CLI (Command Line Interface)

```
sass source/stylesheets/index.scss build/stylesheets/index.css
```

SASS - invocato da CLI

```
c:> sass
Compile Sass to CSS.

Usage: sass <input> [output]

    --[no-]stdin           Read the stylesheet from stdin.
    --[no-]indented        Use the indented syntax for input from stdin.
-I, --load-path=<PATH>     A path to use when resolving imports.
                           May be passed multiple times.

-s, --style=<NAME>         Output style.
                           [expanded (default), compressed]

-c, --[no-]color           Whether to emit terminal colors.
-q, --[no-]quiet           Don't print warnings.
    --[no-]trace           Print full Dart stack traces for exceptions.
-h, --help                 Print this usage information.
    --version              Print the version of Dart Sass.
```

https://github.com/matteobaccan/CorsoSASS

SASS - le variabili

Le variabili ci permettono di definire dei valori che possono essere riutilizzati all'interno del nostro foglio di stile.

Le variabili devono iniziare con il simbolo `$` e possono essere di qualsiasi tipo.

```
$primary-color: #333;

body {
  color: $primary-color;
}
```

SASS - le variabili scope

Le variabili possono essere definite in due modi:

- Globali
- Locali

Le variabili globali possono essere usate in tutto il foglio di stile, mentre le variabili locali possono essere usate solo all'interno del blocco in cui sono definite.

SASS - esempio di scope

La variabile `colore` usata nel selector `h1` è locale, mentre quella usata nel selector `p` è globale.

```
$colore: red;
h1 {
  $colore: green;
  color: $colore;
}
p {
  color: $colore;
}
```

SASS - forzatura dello scope

In questo caso forziamo l'assegnazione di una variabile globale all'interno di un blocco locale.

```
$colore: red;
h1 {
  $colore: green !global;
  color: $colore;
}
p {
  color: $colore;
}
```

SASS - annidamento (nesting)

SASS ci permette di annidare i nostri selettori, in modo da rendere più leggibile il nostro codice.

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li { display: inline-block; }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

SASS - proprietà annidate

SAAS ci permette di annidare anche le proprietà, per maggior chiarezza del codice.

```
.line {
   font: {
      family: Lucida Sans Unicode;
      size:20px;
      weight: bold;
   }
}
```

CSS

```
.line {
  font-family: Lucida Sans Unicode;
  font-size: 20px;
  font-weight: bold;
}
```

SASS - @Import

La direttiva `@import` ci permette di importare un file SASS all'interno di un altro file SASS.

```
@import 'variables';
```

La differenza fra `@import` di SASS, rispetto a quella di CSS, è che SASS importa il file e lo compila all'interno del file in cui è stato importato, mentre CSS importa il file e lo mantiene separato.

Questa differenza è molto importante, perché ci permette di creare un unico file CSS, invece di avere più file CSS.

A livello prestazioni, questo è un vantaggio, perché il browser deve scaricare un solo file CSS, invece di più file CSS.

SASS - @mixin

La direttiva `@mixin` ci permette di creare dei blocchi di codice che possono essere riutilizzati all'interno del nostro foglio di stile.

```
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
     -moz-border-radius: $radius;
      -ms-border-radius: $radius;
          border-radius: $radius;
}

.box {
   @include border-radius(10px);
}
```

SASS - @extend

La direttiva `@extend` ci permette di estendere un selettore con le proprietà di un altro selettore.

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

.seriousError {
  @extend .error;
  border-width: 3px;
}
```

SASS - i commenti

SASS ci permette di usare i commenti in due modi:

- Commenti in linea
- Commenti multilinea

```
// Questo è un commento in linea
/* Questo è un commento
   multilinea */
```

SASS - le funzioni

SASS ci permette di creare delle funzioni che possono essere riutilizzate all'interno del nostro foglio di stile. In questo modo SASS diventa un vero e proprio linguaggio di programmazione.

```
@function pow($base, $exponent) {
  $result: 1;
  @for $_ from 1 through $exponent {
    $result: $result * $base;
  }
  @return $result;
}
```

SASS - tipologie di funzioni

Esistono una serie di funzioni predeterminate da SASS, che ci permettono di manipolare i nostri valori.

- Funzioni per la manipolazione delle stringhe
- Funzioni per la manipolazione dei colori
- Funzioni per la manipolazione delle liste
- Funzioni per la manipolazione dei numeri
- Funzioni per la manipolazione dei mappe

Fonti usate per la creazione di queste slide

https://it.wikipedia.org : definizioni e argomenti

https://www.w3schools.com/sass/ : SASS tutorial

https://sass-lang.com/ : Sito ufficiale di SASS

https://chat.openai.com : ChatGPT

Ogni immagine inserita riporta la fonte

Disclaimer

L'autore ha generato questo testo in parte con GPT-3, il modello di generazione del linguaggio su larga scala di OpenAI. Dopo aver generato la bozza della lingua, l'autore ha rivisto, modificato e rivisto la lingua a proprio piacimento e si assume la responsabilità ultima del contenuto di questa pubblicazione.