

YASS

Yet Another cSS

Xhorxho Papallazi, Marco Rodolfi, Simone Ronzoni

UniBG

February 28, 2024

Yet **A**nother c**SS**

Yet Another cSS

CSS "programmer first":

- preprocessore CSS
- sintassi snella
- funzionalità aggiuntive:
 - Variabili
 - Cicli
 - Mixin
 - Annidamenti

Variabili

Permettono il riutilizzo di valori all'interno dell'intero foglio di stile.

```
1 primary-color = "blue";
2 colors = [ ${primary-color}, "red"];
3 font-sizes = {
4   "small": 0.5em,
5   "normal": 1em,
6   "big": 2em
7 };
8 buttons = {
9   "primary": ${primary-color},
10  "secondary": "gray"
11 };
```

Cicli

Permettono la generazione dinamica di codice CSS

```
1 $foreach(colors) {
2   .color-#{index} {
3     color: #{value};
4   }
5 }
6 $foreach(button-name, button-color: buttons) {
7   $foreach(font-name, font-size: font-sizes) {
8     .btn-#{button-name}_#{font-name} {
9       $button(button-color, font-size);
10      &:hover {
11        border: red;
12        border-size: 2px;
13      }
14    }
15  }
16 }
```

Mixin

Permettono di creare blocchi di codice riutilizzabili

```
1 button = (color, font-size) {  
2   color: ${color};  
3   font-size: ${font-size};  
4 };
```

```
1   $button(button-color, font-size);
```

Annidamenti

Migliorano la leggibilità e manutenibilità del codice

```
1      .btn-${button-name}__${font-name} {  
2          $button(button-color, font-size);  
3          &:hover {  
4              border: red;  
5              border-size: 2px;  
6          }  
7      }
```

YASS vs CSS

```
1 primary-color = "blue";
2 colors = [ ${primary-color}, "red"];
3 font-sizes = {
4   "small": 0.5em,
5   "normal": 1em,
6   "big": 2em
7 };
8 buttons = {
9   "primary": ${primary-color},
10  "secondary": "gray"
11 };
12 button = (color, font-size) {
13   color: ${color};
14   font-size: ${font-size};
15 };
16 $foreach(colors) {
17   .color-${index} {
18     color: ${value};
19   }
20 }
21 $foreach(button-name, button-color: buttons)
22   {
23     $foreach(font-name, font-size: font-sizes)
24     {
25       .btn-${button-name}__${font-name} {
26         $button(button-color, font-size);
27         &:hover {
28           border: red;
29           border-size: 2px;
30         }
31       }
32     }
33   }
```

```
1 .color-0 {
2   color: blue;
3 }
4 .color-1 {
5   color: red;
6 }
7 .btn-primary__small {
8   color: blue;
9   font-size: 0.5em;
10 }
11 .btn-primary__small:hover {
12   border: red;
13   border-size: 2px;
14 }
15 .btn-primary__normal {
16   color: blue;
17   font-size: 1em;
18 }
19 .btn-primary__normal:hover {
20   border: red;
21   border-size: 2px;
22 }
23 .btn-primary__big {
24   color: blue;
25   font-size: 2em;
26 }
27 .btn-primary__big:hover {
28   border: red;
29   border-size: 2px;
30 }
31 ... (23 lines more)
```


Come si installa?

Partire installando Java e scaricare il file jar, fine...

- tutte le dipendenze sono incluse
- nessuna configurazione è necessaria
- integrabile nella pipeline di sviluppo

Come si usa?

- Scrivere un file in formato `.yass`
- Compilarlo con `java -jar yass.jar <input>.yass <output>.css`
- Utilizzare il file compilato `<output>.css` nella pagina HTML

Alternative già esistenti

Esistono già alternative sul mercato, come SASS, Less, Stylus e PostCSS. YASS fornisce le funzionalità più necessarie mantenendo una sintassi familiare ai programmatori!

YASS vs SASS - Vantaggi

- Sintassi semplice, intuitiva, coerente, snella.
- Java è già preinstallato sulla maggior parte dei computer, quindi non è necessario gestire l'installazione di npm

YASS vs SASS - Vantaggi

```
1 primary-color = "blue";
2 colors = [ ${primary-color}, "red"];
3 font-sizes = {
4   "small": 0.5em,
5   "normal": 1em,
6   "big": 2em
7 };
8 buttons = {
9   "primary": ${primary-color},
10  "secondary": "gray"
11 };
12 button = (color, font-size) {
13   color: ${color};
14   font-size: ${font-size};
15 };
16 $foreach(colors) {
17   .color-${index} {
18     color: ${value};
19   }
20 }
21 $foreach(button-name, button-color: buttons)
22 {
23   $foreach(font-name, font-size: font-sizes)
24   {
25     .btn-${button-name}__${font-name} {
26       $button(button-color, font-size);
27       &:hover {
28         border: red;
29         border-size: 2px;
30       }
31     }
32   }
```

```
1 $primary-color: blue;
2 $colors: $primary-color, red;
3 $font-sizes: (
4   "small": 0.5em,
5   "normal": 1em,
6   "big": 2em);
7 $buttons: (
8   "primary": $primary-color,
9   "secondary": gray);
10 @mixin button($color, $font-size) {
11   color: $color;
12   font-size: $font-size;
13 }
14 @for $i from 1 through length($colors) {
15   $color: nth($colors, $i);
16   .color-#{ $i } {
17     color : $color;
18   }
19 }
20 @each $button-name, $button-color in $buttons
21 {
22   @each $font-name, $font-size in $font-sizes
23   {
24     .btn-#{ $button-name }__#{ $font-name } {
25       @include button($button-color, $font-
26         size);
27       &:hover {
28         border: red ;
29         border-size: 2px;
30       }
31     }
32   }
```

YASS vs SASS - Svantaggi

- YASS implementa meno funzionalità rispetto a SASS (ad esempio cicli nei mixin, liste di tuple, dizionari di dizionari, etc.)
- YASS è una tecnologia nuova quindi non diffusa come SASS

Conclusione

- YASS ha una curva di apprendimento molto veloce per programmatori che arrivano da linguaggi strutturati
- Anche aggiungendo ulteriori funzionalità, la sintassi rimarrà semplice