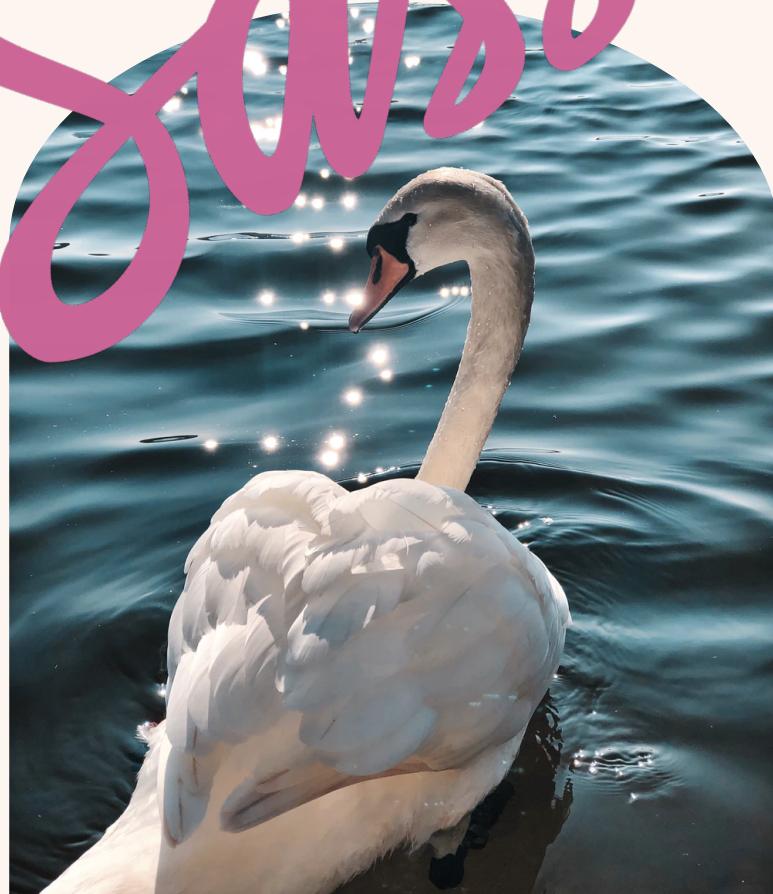


Sassy





Syntactically Awesome Style Sheets

CSS를 편리하게 이용할 수 있도록 도와주며

추가 기능도 있는 확장판 (CSS를 확장하는 스크립팅언어)

SCSS

nesting(중첩)

| 셀렉터 네스팅

html 구조상 div 안에 있는 p 처럼 하위구조를 가지고 있을 경우, div의 속성 정의 블록 안에 하위에 있는 p가 들어갈 수 있다. scss 파일 만으로도 html 구조를 알 수 있으며 타고 들어가는 것이 많을 때 유용하다.

```
// SCSS
.div {
  p {
    color:skyblue;
  }
}
```

```
// CSS
.div p
  {color:skyblue}
```

| 속성 네스팅

축약형으로 묶일 수 있는 속성들끼리도 네스팅이 가능하다. 예를 들어 font의 경우font-family, font-size, font-weight등이 있는데 다음과 같이 속성의 하위 사전 형태로 작성할 수 있다.

```
.entry-content {
  p {
    font: {
      family: "Noto Serif CJK KR", serif;
      size: 9.814rem;
      weight: 400;
    }
  }
}
```

@mixin & @include

| @mixin

믹스인을 사용하면 스타일시트 전체에서 재사용할 수 있는 스타일을 정의할 수 있다. 믹스인은 다른 규칙에 내포하거나 스타일 시트의 최상위 수준에 포함할 수 있는 스타일 규칙을 포함할 수도 있다.

| @include

@include는 @mixin을 불러올 때 사용함으로서 함께 하는 짹꿍이라고 볼 수 있다.

// 믹스인 사용했을 때

```
@mixin reset-list {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav ul {  
  @include reset-list;  
}  
ul li {  
  @include reset-list;  
}
```

// 믹스인 사용하지 않았을 때

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
ul li {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}
```

SCSS for

기본적으로 Sass의 다른 문법 및 함수 사용과 마찬가지로 @(골뱅이) 문자와 함께 for문을 사용한다. 각 반복마다 카운트 변수를 사용할 수 있다. **@for 변수 from (시작 숫자) to (끝 숫자){...}** 의 형태로 작성하고 to 대신 through를 작성할 수 있다.

```
/* SCSS */
@for $i from 0 to 5{
  .animal_#{$i}{
    background-position:0 ($i * -10px);
  }
}
/* CSS */
.animal_0{background-position:0 0;}
.animal_1{background-position:0 -10px;}
.animal_2{background-position:0 -20px;}
.animal_3{background-position:0 -30px;}
.animal_4{background-position:0 -40px;}
```

to 뒤에 나오는 숫자 미만의 반복되는 값을 사용한다.
(해당 숫자 비포함)

```
/* SCSS */
@for $i from 0 through 5{
  .animal_#{$i}{
    background-position:0 ($i * -10px);
  }
}
/* CSS */
.animal_0{background-position:0 0;}
.animal_1{background-position:0 -10px;}
.animal_2{background-position:0 -20px;}
.animal_3{background-position:0 -30px;}
.animal_4{background-position:0 -40px;}
.animal_5{background-position:0 -0px;}
```

through 뒤에 나오는 숫자 이하로 반복되는 값을 사용한다.
(해당 숫자 포함)

Interpolation

보간법(Interpolation)

Sass 스타일시트의 거의 모든 위치에서 보간법을 사용할 수 있습니다. \${}

- * 스타일 규칙의 선택자 (Selectors in style rules)
- * 속성 이름 (Property names in declarations)
- * 사용자 지정 속성 값 (Custom property values)
- * CSS at-rules
- * @extend
- * 일반 CSS @import
- * 인용된 문자열 및 인용되지 않은 문자열 (Quoted or unquoted strings)
- * 함수 (Special functions)
- * 일반 CSS 함수 이름 (Plain CSS function names)
- * 주석 (Loud comments)

//SCSS

```
@ mixin corner-icon($name, $top-or-bottom, $left-or-right) {  
  .icon-#${$name} {  
    background-image: url("/icons/#{$name}.svg");  
    position: absolute;  
    #{$top-or-bottom}: 0;  
    #{$left-or-right}: 0;  
  }  
}  
  
@include corner-icon("mail", top, left);
```

// CSS

```
.icon-mail {  
  background-image: url("/icons/mail.svg");  
  position: absolute;  
  top: 0;  
  left: 0;  
}
```

SCSS

변수(variable)

sass 변수는 \$로 시작하는 이름에 값을 할당한 다음 값 자체 대신 해당 이름을 참조할 수 있다. 변수를 사용하면 반복을 줄이고, 복잡한 계산을 수행하며, 라이브러리를 구성하는 등의 작업을 수행할 수 있다. **\$variable-name: variable-value;**

//SCSS

```
$font-main : 'Six Caps';
$gradient1 : linear-gradient(to bottom, #11e8bb
0%,#8200c9 100%);
//그라디언트도 변수적용이 가능하다.

main{
  background : $gradient1;
  font family : font-main;
}
```

// CSS

```
.main {
  background : linear-gradient(to bottom, #11e8bb
0%,#8200c9 100%);
  font family : 'Six Caps';
}
```