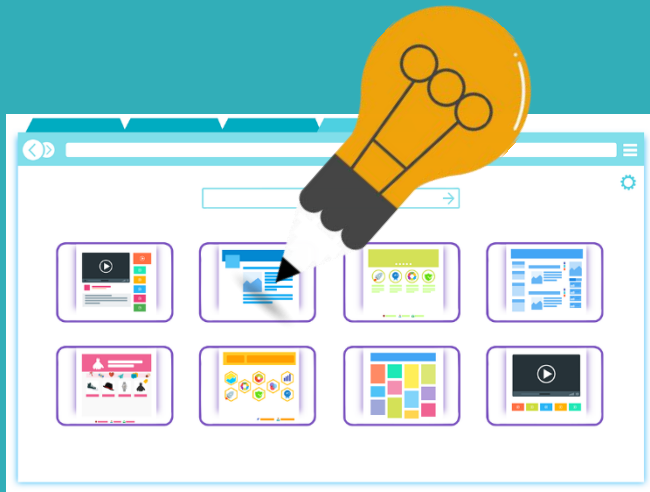


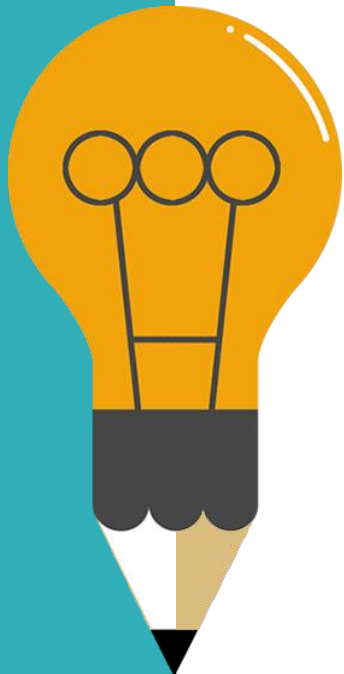
※ 재 배포 및 편집은 금지됩니다.(저작권 관련)



CSS 라이브러리 소개

HTML과 CSS를 더 빠르고 쉽게 관리하는 방법

배울 내용은...



01

Emmet

02

유용한 tip- calc 함수, 커스텀 속성

03

Sass

04

폰트 표시 단위

05


svg

06

css 라이브러리


https://emmet.io/

- 에밋으로 빠르게 코딩하기




Emmet

Tools for web-developers

[DOCUMENTATION](#) 

[DOWNLOAD](#)

[CREDITS](#)

 [ads via Carbon](#)

[Your AD here](#)
[ads via Carbon](#)


[Become a sponsor](#)
[or backer](#)

Sponsors:

Emmet — the essential toolkit for web-developers

Emmet is a plugin for many popular text editors which greatly improves HTML & CSS workflow:

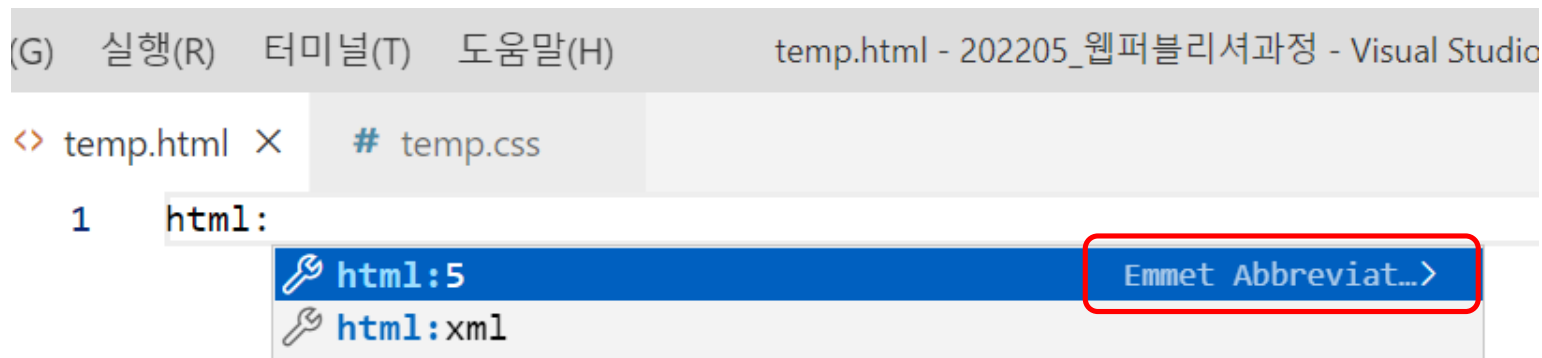
```
<!doctype html>
<html lang="en">
<head>
  <title>Demo</title>
</head>
<body>
</body>
</html>
```



Watch demo

https://emmet.io/

- html 코딩



!

doc

<https://emmet.io/>

■ html 코딩

사용법		예시
태그 이름	+ [tab키]	p
태그 이름 + class 이름		p.text
태그 이름 + id 이름		p#text
자식 요소가 있는 태그		ul>li
여러 태그 반복 입력		ul>li*3
같은 계층 입력		div+p
한 단계 상위계층 입력		ul>li^div
텍스트 입력		p{텍스트}
일련번호 붙이기		li.menu\$*3

<https://emmet.io/>

■ CSS 코딩

사용법			자동완성
p{	m	+ [tab키] 또는 [enter키]	margin: ;
	mb		margin-bottom: ;
	w100		width: 100px;
	w100p		width: 100%;
	w100r		width: 100rem;
	w100e		width: 100em;
	c#0bd		color: #0bd;
	m10-20-15-20		margin: 10px 20px 15px 20px;

<https://emmet.io/>

■ CSS 코딩

사용법			자동완성
p{	h	+ [tab키] 또는 [enter키]	height: ;
	fz		font-size: ;
	bg		background:#000;
	bd		border:1px solid #000;
	df		display:flex;
	ta		text-align:left;

<https://docs.emmet.io/cheat-sheet/>

- 그 외의 입력 방법은

[← Emmet Documentation](#)

Download cheat sheet as printable PDF A5

Your AD here
ads via Carbon

Syntax

Child: >

```
nav>ul>li
<nav>
  <ul>
    <li></li>
  </ul>
</nav>
```

Sibling: +

```
div+p+bq
<div></div>
<p></p>
<blockquote></blockquote>
```

Climb-up: ^

```
div+div>p>span+em^bq
<div></div>
<div>
  <p><span></span></span><em></em></p>
```

Multiplication: *

```
ul>li*5
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

Item numbering: \$

```
ul>li.item$*5
<ul>
  <li class="item1"></li>
  <li class="item2"></li>
  <li class="item3"></li>
  <li class="item4"></li>
  <li class="item5"></li>
</ul>
```

ID and CLASS attributes

```
#header
<div id="header"></div>

.title
<div class="title"></div>

form#search.wide
<form id="search" class="wide"></form>

p.class1.class2.class3
<p class="class1 class2 class3"></p>
```

Custom attributes

```
p[title="Hello world"]
<p title="Hello world"></p>

td[rowspan=2 colspan=3 title]
<td rowspan="2" colspan="3" title=">
```


calc 함수로 계산식 사용하는 법

- 크기를 정할 때 사용

```
div {  
  background: #0bd;  
  width: calc(100% / 3);  
  height: 100px;  
  padding: 16px;  
}
```

※

연산자 앞 뒤에 꼭 공백 넣기

사용 가능 연산자: + - * /

calc 함수로 계산식 사용하는 법

- 계산 단위가 달라도 사용 가능

```
div {  
    background: #0bd;  
    height: calc(100vh - 100px);  
    padding: 16px;  
}
```

calc 함수로 계산식 사용하는 법

- 커스텀 속성(변수)과 조합

```
div {  
  --number: 500;  
  width: calc(var(--number) * 1px);  
  background: #0bd;  
  padding: 16px;  
}
```

커스텀 속성

- CSS변수, 캐스케이드 변수

```
a{
  color:#4db1ec;
}
.btn{
  background-color: #4db1ec;
}
.border{
  border: 1px solid #4db1ec;
}
.title{
  color: #4db1ec;
}
```

```
:root {
  --main-color: #4db1ec;
}
a{
  color:var(--main-color);
}
.btn{
  background-color: var(--main-color);
}
.border{
  border: 1px solid var(--main-color);
}
.title{
  color: var(--main-color);
}
```

커스텀 속성 주의점

- 대소문자 구분함
- 커스텀 속성에 속성명을 넣을 수는 없다.
- 커스텀 속성을 호출한 뒤 단위를 덧붙이면 적용되지 않는다.

커스텀 속성 주의점

- 대소문자 구분함
- 커스텀 속성에 속성명을 넣을 수는 없다.
- 커스텀 속성을 호출한 뒤 단위를 덧붙이면 적용되지 않는다.

```
:root{  
  --bg: pink;  
  --BG: orange;  
}
```

var(-bg)

var(-BG)

커스텀 속성 주의점

- 대소문자 구분함
- 커스텀 속성에 속성명을 넣을 수는 없다.
- 커스텀 속성을 호출한 뒤 단위를 덧붙이면 적용되지 않는다.

```
p{  
  --bg: background-color;  
  var(--bg): pink;  
}
```

```
p{  
  --bg: pink;  
  background-color: var(--bg);  
}
```

커스텀 속성 주의점

- 대소문자 구분함
- 커스텀 속성에 속성명을 넣을 수는 없다.
- 커스텀 속성을 호출한 뒤 단위를 덧붙이면 적용되지 않는다.

```
div{  
  --main-padding: 2;  
  padding: var(--main-padding)rem;  
}
```

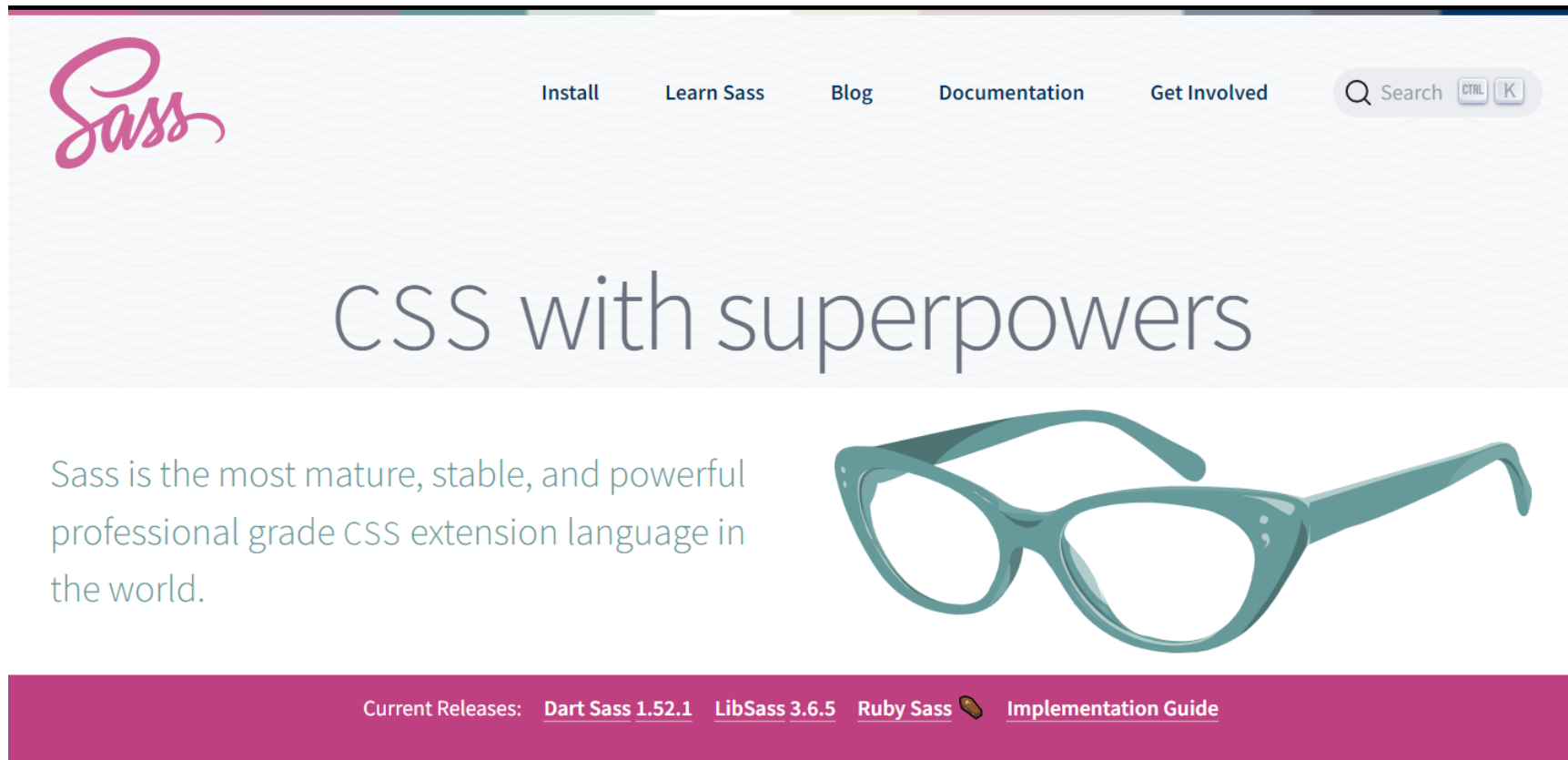
```
div{  
  --main-padding: 2rem;  
  padding: var(--main-padding);  
}
```


커스텀 속성 주의점

- calc 함수 사용하면 단위를 빼고도 사용 가능

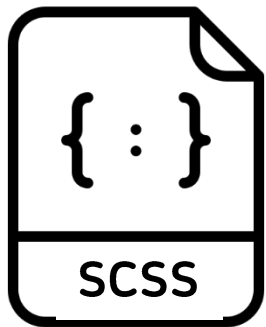
```
div{  
  --main-padding: 2;  
  padding: calc(var(--main-padding) * 1rem );  
}
```

Sass - <https://www.sass-lang.com/>



Sass

- CSS를 보조하는 새로운 도구
- 파일 확장자: .scss, .sass



style.scss 작성



변환
(컴파일)



style.css

Sass 장점

- 셀렉터 부모 자식 관계를 네스트로 구현
- 변수로 값 다시 사용하기
- 파일을 분할해서 관리

```
.nav{  
  padding: 10px;  
  
  ul{  
    list-style:none;  
  
    a{  
      color:#0bd;  
    }  
  }  
}
```

```
1 .nav {  
2   padding: 10px;  
3 }  
4 .nav ul {  
5   list-style: none;  
6 }  
7 .nav ul a {  
8   color: #0bd;  
9 }
```

Sass 장점

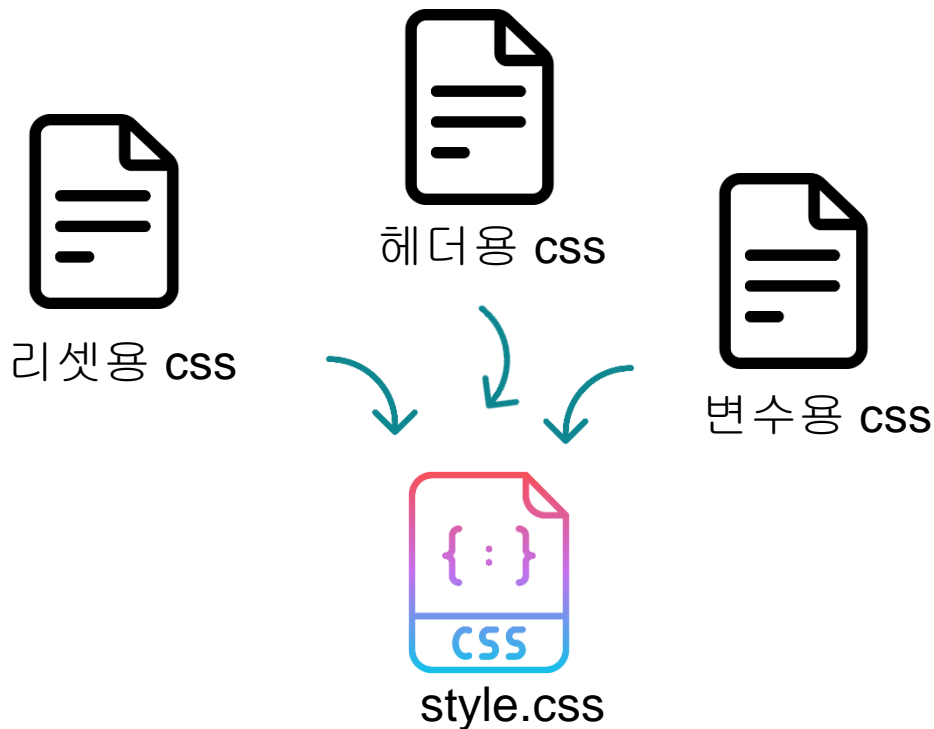
- 셀렉터 부모 자식 관계를 네스트로 구현
- 변수로 값 다시 사용하기
- 파일을 분할해서 관리

```
$base-color: #ddd;  
$main-color: #0bd;  
  
body{  
    background: $base-color;  
}  
a{  
    color: $main-color;  
}
```


```
1 body {  
2     background: #ddd;  
3 }  
4  
5 a {  
6     color: #0bd;  
7 }
```

Sass 장점

- 셀렉터 부모 자식 관계를 네스트로 구현
- 변수로 값 다시 사용하기
- 파일을 분할해서 관리



Sass 체험하기 - SassMeister

FileViewSassOptionsLog in with your GitHub account to save gists

SCSS

dart-sass v1.32.12

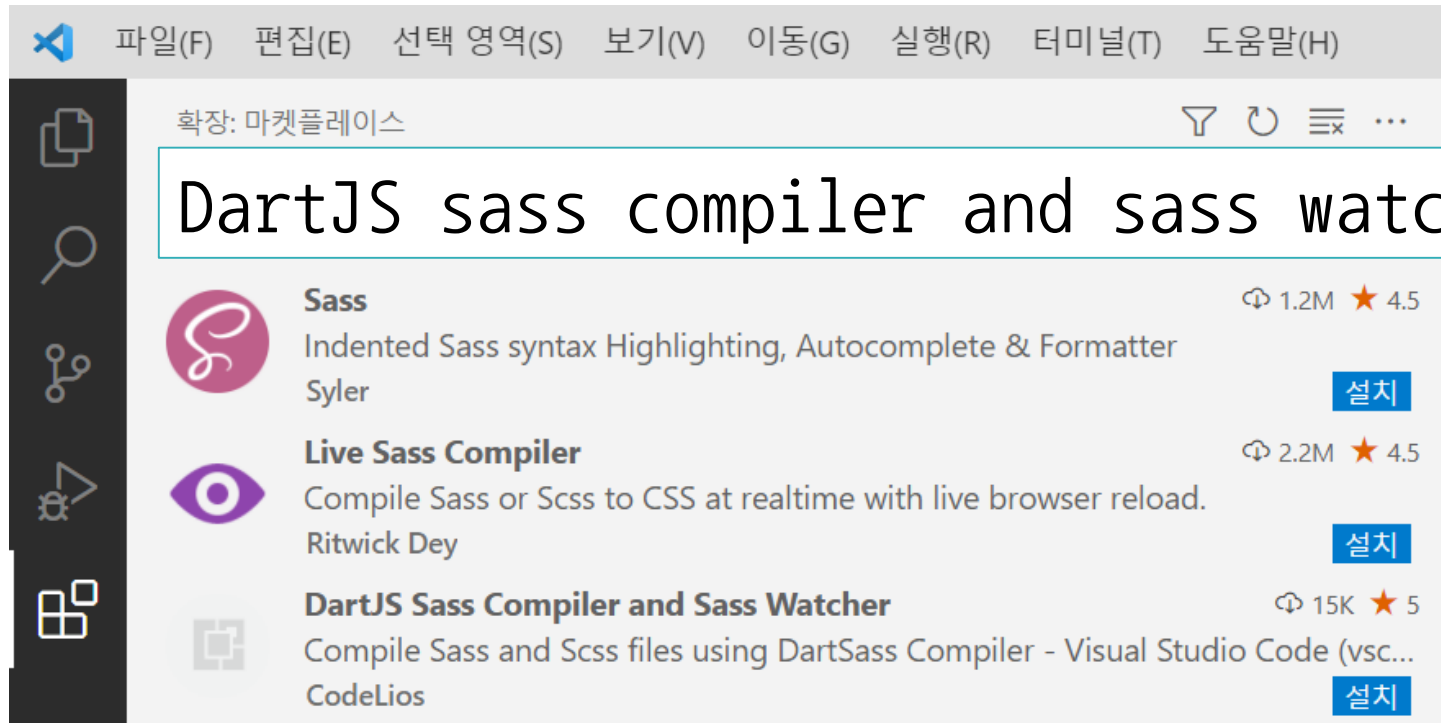
```
1 .nav{
2   padding: 10px;
3
4   ul{
5     list-style:none;
6
7     a{
8       color:#0bd;
9     }
10  }
11 }
```

CSS

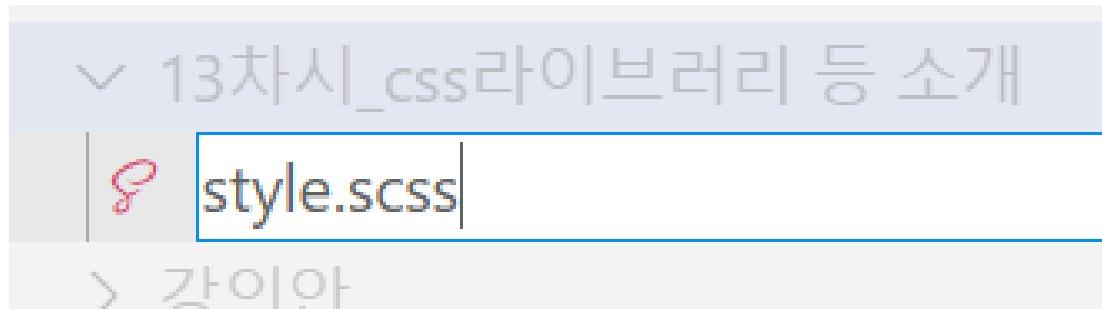
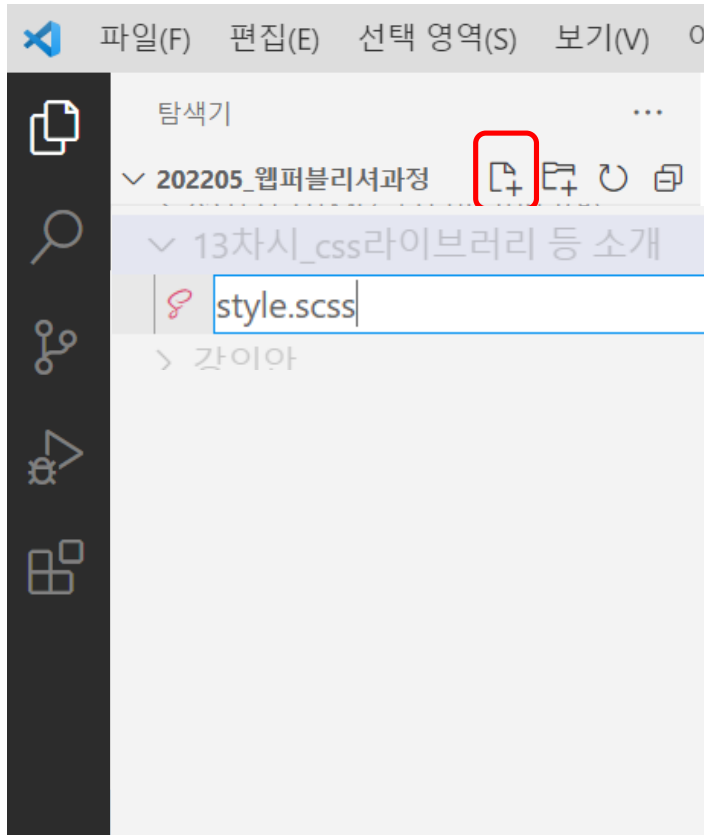
```
1 .nav {
2   padding: 10px;
3 }
4 .nav ul {
5   list-style: none;
6 }
7 .nav ul a {
8   color: #0bd;
9 }
```

Sass - VS code에서 사용하기

■ 확장 기능 설치하기



Sass - 내 작업 폴더에서 파일 생성



Sass - VS code에서 사용하기

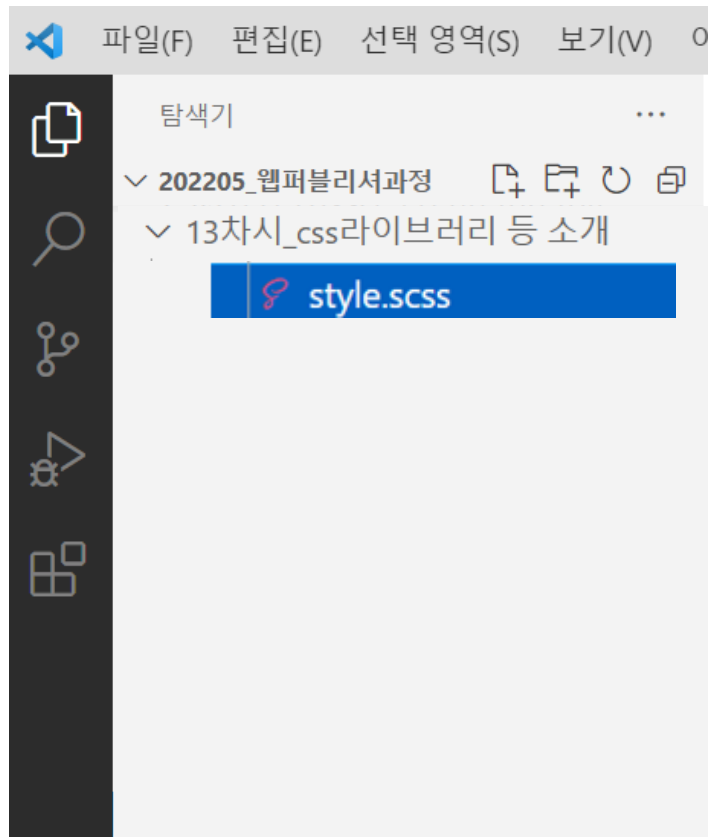


A screenshot of the Visual Studio Code (VS Code) editor interface. The top menu bar shows options: 파일(F), 편집(E), 선택 영역(S), 보기(V), 이동(G), 실행(R), and 터미널. The left sidebar contains icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The main editor area displays a file named 'style.scss' with the following Sass code:

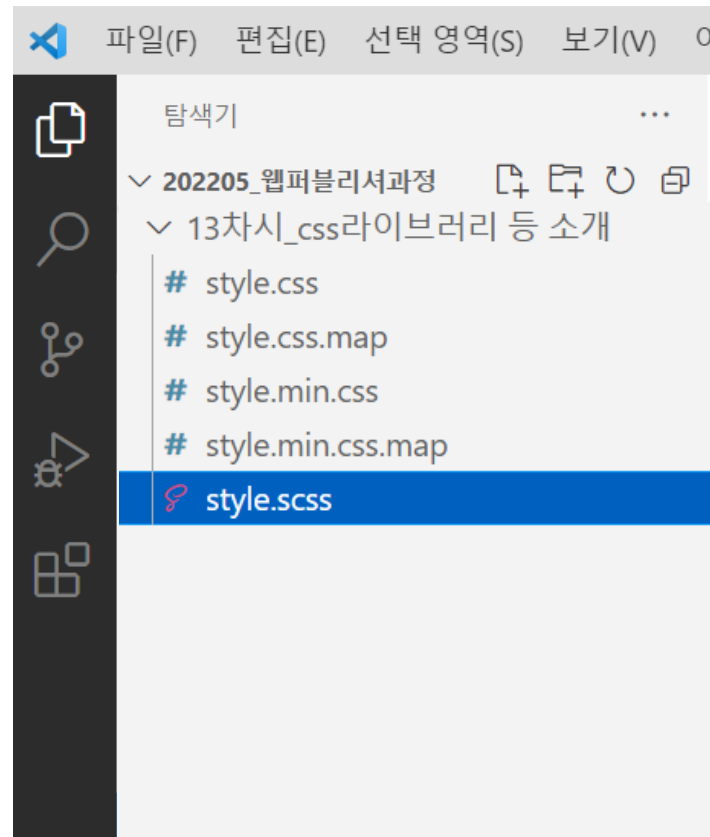
```
1 $main-color: #0bd;  
  
    .nav {  
        padding: 10px;  
  
        a {  
            color: $main-color;  
        }  
    }
```



Sass - VS code에서 사용하기



저장하면 자동
으로 **css** 파일
생성됨



Sass - VS code에서 사용하기

style.css

style.css.map

style.min.css

style.min.css.map

🔗 style.scss

변환된 CSS 파일

CSS 파일을 합치기 위한 소스맵

들여쓰기, 줄바꿈, 주석이 모두 제외된 파일

SCSS 파일

Sass 실습

- 네스트 활용하기
 - &를 붙여서 부모 셀렉터 참고하기
 - 클래스명도 네스트 가능
 - 속성명도 네스트 사용

Sass 실습

- 과도한 네스트 사용 주의

```
.main-menu{  
  margin: 10px;  
  nav{  
    background: #ddd;  
    ul{  
      display: flex;  
      li{  
        margin: 5px;  
        a{  
          color: #0bd;  
          span{  
            font-size: .85rem;  
          }  
        }  
      }  
    }  
  }  
}
```

나쁜 예

Sass 실습

- 과도한 네스트 사용 주의

```
.main-menu{  
  margin: 10px;  
  nav{  
    background: #ddd;  
  }  
  ul{  
    display: flex;  
  }  
  li{  
    margin: 5px;  
  }  
  a{  
    color: #0bd;  
  }  
  span{  
    font-size: .85rem;  
  }  
} // .main-menu
```

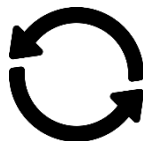
좋은 예

파셜 파일로 분할하기

- patial : 분할된 파일



_header.scss



_header.scss 내용

style.scss 내용

@use 'header';



style.scss

mixin으로 스타일 재사용하기

- mixin: 미리 저장한 스타일을 필요한 곳에서 불러와 재사용하는 기능

```
@mixin circle {  
    width: 80px;  
    height: 80px;  
    border-radius: 50%;  
}
```

```
div { @include circle;}
```

mixin으로 스타일 재사용하기

- 인수 사용하기: 전해지는 값
 - 속성 값이 들어가는 부분에 변수로 처리

```
@mixin circle($size){  
  width: $size;  
  height: $size;  
  border-radius: 50%;  
}
```

```
div {  
  @include circle(60px);  
}
```

mixin으로 스타일 재사용하기

- 인수 사용하기: 여러 개도 사용 가능

```
@mixin circle($size, $bg){  
  width: $size;  
  height: $size;  
  background: $bg;  
  border-radius: 50%;  
}
```

```
div {  
  @include circle(60px, #0bd);  
}
```

mixin으로 스타일 재사용하기

- 초깃값 설정

```
@mixin circle($size:80px){  
  width: $size;  
  height: $size;  
  border-radius: 50%;  
}
```

```
div {  
  @include circle;  
}  
img {  
  @include circle(100px);  
}
```

폰트 표시 단위

- px : 고정 크기

```
<style>
    .wrapper {
        font-size:16px;
    }
</style>
<body>
    <div class="wrapper">
        <div>16px 크기의 텍스트입니다.</div>
    </div>
</body>
```

폰트 표시 단위

- em : 상대 크기(기준: 부모요소)

```
<style>
    .wrapper {
        font-size:16px;
    }
    h1 { font-size:3em; }
    p { font-size:1.5em;}
</style>
<body>
    <div class="wrapper">
        <h1>3em 크기의 제목 </h1>
        <p>1.5em 크기의 텍스트입니다.</p>
        <div>16px 크기의 텍스트입니다.</div>
    </div>
</body>
```

3em 크기의 제목

1.5em 크기의 텍스트입니다.

16px 크기의 텍스트입니다.

폰트 표시 단위

- em : 상대 크기(기준: 부모요소)

```
<style>
.wrapper { font-size:16px;}
div { font-size:1.5em; }
</style>
```

```
<body>
```

```
  <div class="wrapper">
```

```
    <div>1.5em 크기의 텍스트입니다.
```

```
      <div>1.5em 크기의 텍스트입니다.
```

```
        <div>1.5em 크기의 텍스트입니다.</div>
```

```
      </div>
```

```
    </div>
```

```
  </div> </body>
```

1.5em 크기의 텍스트입니다.

1.5em 크기의 텍스트입니다.

1.5em 크기의 텍스트입니다.

폰트 표시 단위

- rem: Root em

```
<style>
  .wrapper { font-size:16px;}
  div { font-size:1.5rem; }
</style>
```

```
<body>
  <div class="wrapper">
    <div>1.5em 크기의 텍스트입니다.
      <div>1.5em 크기의 텍스트입니다.
        <div>1.5em 크기의 텍스트입니다.</div>
      </div>
    </div>
  </div> </body>
```

1.5rem 크기의 텍스트입니다.
1.5rem 크기의 텍스트입니다.
1.5rem 크기의 텍스트입니다.

SVG

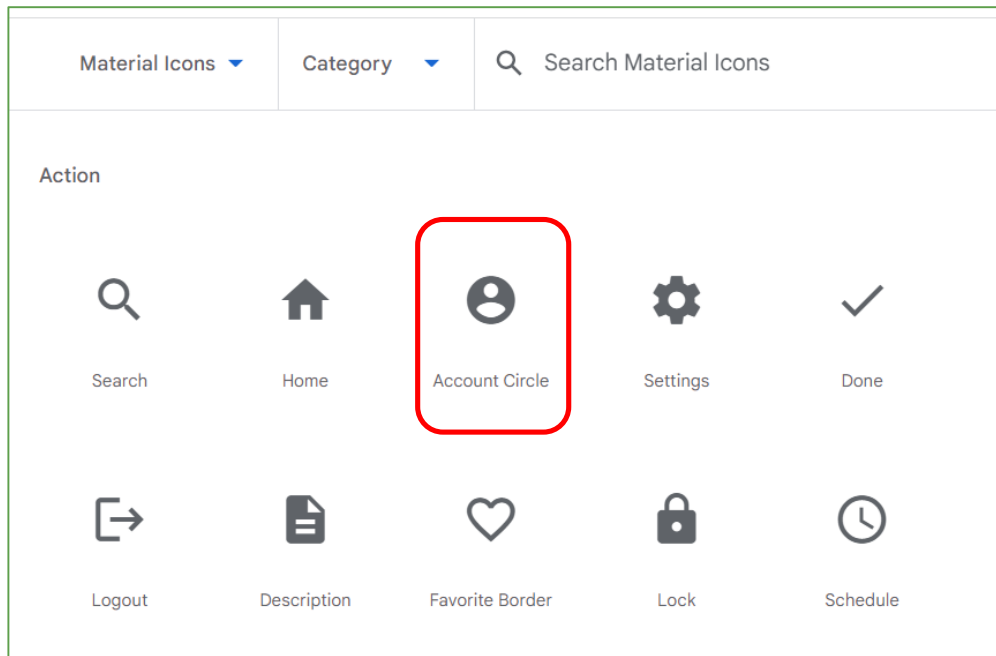
- 화질에 영향 받지 않는 벡터 이미지
- 스타일 수정 용이
- xml 기반의 문서
- 애니메이션, css3 효과 적용 가능

SVG

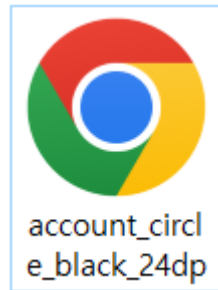
- 어디에 사용?
 - 로고와 아이콘
 - 데이터 시각화(시각화 프레임워크 사용)

<https://material.io/icons>

- 머티리얼 SVG 아이콘 다운로드



diskette



SVG 사용하기

- 태그
- SVG 소스

```

```

```
<svg xmlns="http://www.w3.org/2000/svg" enable-background="new 0 0 24  
24" height="24px" viewBox="0 0 24 24" width="24px"  
fill="#000000"><g><rect fill="none" height="24"  
width="24"/></g><g><path d="M12,2C6.48,2,2,6.48,2,12s4.48,10,10,10s10-  
4.48,10-10S17.52,2,12,2z M12,6c1.93,0,3.5,1.57,3.5,3.5S13.93,13,12,13  
s-3.5-1.57-3.5-3.5S10.07,6,12,6z M12,20c-2.03,0-4.43-0.82-6.14-  
2.88C7.55,15.8,9.68,15,12,15s4.45,0.8,6.14,2.12  
C16.43,19.18,14.03,20,12,20z"/></g></svg>
```

CSS 라이브러리



Animate.CSS

- 사용하기 쉬운 CSS 애니메이션 라이브러리
- <https://animate.style/>

imagehover CSS

- <https://imagehover.io/>



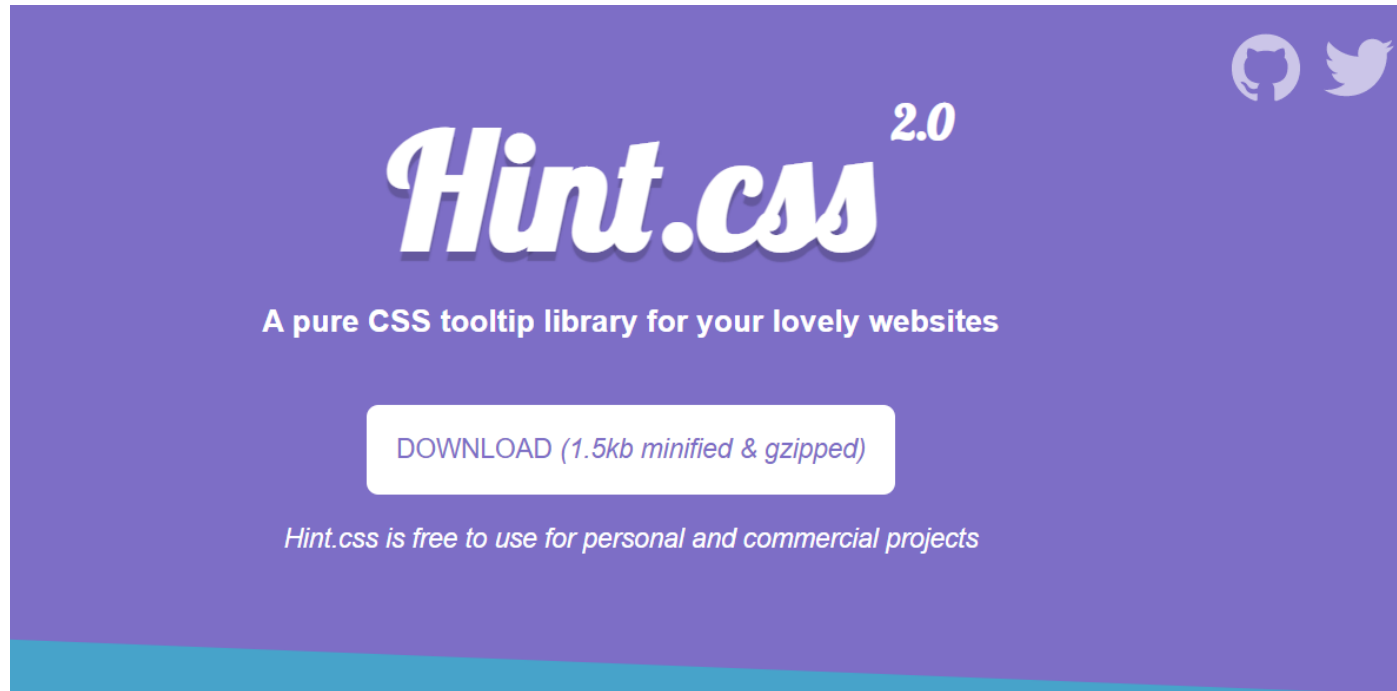
loading.io

- 로딩 화면, 로딩 프로그래스

The screenshot displays the loading.io website interface. At the top, the navigation bar includes 'LOADING.IO', 'Upload', 'Icon', 'Text', 'Background', 'Progress Bar', 'Color', 'Libraries', 'Pricing', 'Login', and 'Sign Up'. The main banner features the text 'Animation Made Easy.' with a 'Get Start' button. To the left, there's a promotion for Adobe Stock images. On the right, three sections are visible: 'Icon Library' (animate any icons), 'Your Image' (upload and animate image), and 'Your Words' (make unique text animation). A 'more features ...' link is also present. At the bottom, a configuration bar allows users to customize their animation with color pickers for Color 1 (#e15b64), Color 2 (#f47e60), Color 3 (#f8b26a), and Color 4 (#abbd81). It also includes sliders for Block Size (0 to 49), Strength (1 to 1.15), and a 'Speed' control. A 'TRY IT NOW' starburst is overlaid on the left side of the configuration bar.

<https://kushagra.dev/lab/hint/>

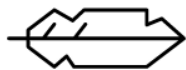
■ 힌트



<http://getskeleton.com/>

A dead simple, responsive boilerplate.

DOWNLOAD



Light as a feather at ~400 lines &
built with mobile in mind.



Styles designed to be a starting
point, not a UI framework.



Quick to start with zero compiling
or installing necessary.

INTRO

CODE

EXAMPLES

MORE
