



# Preprocessors



## Preprocessors

### Fundamentals

- › Preprocessor nedir?
- › SASS
- › LESS
- › Stylus
- › Post css



## Preprocessors

# CSS



Clean



Manageable



Reusable



## Preprocessors

SASS  
LESS  
Stylus  
Post CSS  
...



**CSS**



# Preprocessors

● Sass  
Programming language

● Less  
Stylesheet language

● Stylus  
Topic

● post css  
Search term

+

Worldwide ▼

Past 12 months ▼

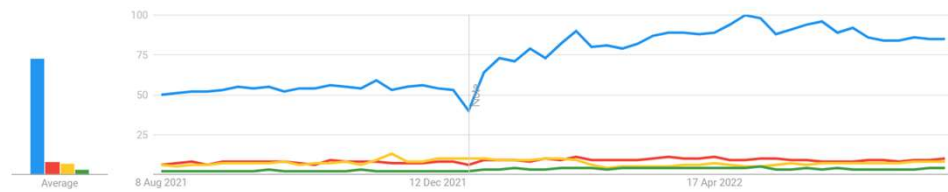
All categories ▼

Web Search ▼

**Note:** This comparison contains both Search terms and Topics, which are measured differently.

[LEARN MORE](#)

Interest over time ⓘ





## Preprocessors

**SASS**

(Syntactically Awesome Style Sheets)

- › Installation
- › Syntax
- › Style Rules
- › Variables
- › Interpolation
- › Modules
- › Mixins
- › Inheritance
- › Conditional Statements
- › Loops



## Installation

SASS, CSS dönüşümü için kullanılabilecek yöntemler



Vscode Extention  
Live Sass Compiler



NPM Package  
`npm i -g sass`



Web Pack

```
sass source:destination  
sass --watch soruce:destination
```



## Syntax

### SCSS

```
h1{  
  text-align:center;  
}  
  
.title{  
  color: red;  
}
```

.SCSS

### SASS

```
h1  
  text-align:center  
  
.title  
  color: red
```

.sass





## Style Rules

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

SCSS

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

CSS



## Style Rules

```
.alert, .warning {  
  ul, p {  
    margin-right: 0;  
    margin-left: 0;  
    padding-bottom: 0;  
  }  
}
```

SCSS

```
.alert ul, .alert p, .warning ul,  
.warning p {  
  margin-right: 0;  
  margin-left: 0;  
  padding-bottom: 0;  
}
```

CSS



## Style Rules

```
ul {  
  > li {  
    list-style-type: none;  
  }  
}
```

SCSS

```
ul > li {  
  list-style-type: none;  
}
```

CSS



## Style Rules

```
h2 {  
  + p {  
    border-top: 1px solid gray;  
  }  
}
```

SCSS

```
h2 + p {  
  border-top: 1px solid gray;  
}
```

CSS



## Style Rules

```
p {  
  ~ span {  
    opacity: 0.8;  
  }  
}
```

SCSS

```
p ~ span {  
  opacity: 0.8;  
}
```

CSS



## Style Rules

```
.alert {  
  color: red;  
  font-weight: bold;  
  text-decoration: none;  
  &:hover {  
    color: green;  
  }  
  &::before{  
    content:"*";  
  }  
}
```

SCSS

```
.alert{  
  color: red;  
  font-weight: bold;  
  text-decoration: none;  
}  
.alert:hover{  
  color: green;  
}  
.alert::before{  
  content:"*"  
}
```

CSS



## Variables

```
$base-color: #c6538c;  
$border-dark: rgba($base-color, 0.88);  
$bs: 0 0.5rem 1rem red;  
  
.alert {  
  border: 1px solid $border-dark;  
  box-shadow: $bs;  
}
```

SCSS

```
.alert {  
  border: 1px solid rgba(198, 83, 140, 0.88);  
  box-shadow: 0 0.5rem 1rem red;  
}
```

CSS



## Interpolation

Bir sass ifadesini sass stilleri içine gömmek için kullanılır. Yapı itibarıyla string literal lere benzer.

```
$name: mail;
$h-position: left;
$v-position: top;

.icon-#{$name} {
  background-image: url("/icons/#{$name}.svg");
  position: absolute;
  #{$h-position}: 0;
  #{$v-position}: 0;
}
```

SCSS

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

CSS





## Modules

Stillerin tek bir dosyada toplanması büyük projelerde sorun oluşturabilir. Bu sebeple proje stilleri farklı dosyalarda tutulabilir. Bu şekilde oluşturulup import edilen her bir dosyaya module denir. Import işlemi için **use** ifadesi kullanılır.



\_variables.scss



\_mixins.scss



\_library.scss

```
@use 'variables';  
@use 'mixins';  
@use 'library';
```

SCSS



## Mixins

Tekrar kullanılabilir ve çağrılabilir stiller oluşturmak için kullanılırlar.

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

SCSS

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

CSS



## Mixins

Mixin ler dışarıdan parametre alarak daha esnek hale getirilebilir.

```
@mixin reset-list($margin, $padding) {  
  margin: $margin;  
  padding: $padding;  
  list-style: none;  
}  
  
nav ul {  
  @include reset-list(5px, 5px);  
}
```

SCSS

```
nav ul {  
  margin: 5px;  
  padding: 5px;  
  list-style: none;  
}
```

CSS



# Inheritance

Bir class in sahip olduğu tüm özellikler, başka bir class a miras olarak aktarılabilir.

```
.button-basic {  
  border: none;  
  padding: 15px 30px;  
  text-align: center;  
  font-size: 16px;  
  cursor: pointer;  
}  
  
.button-report {  
  @extend .button-basic;  
  font-size: 18px;  
}
```

SCSS

```
.button-report {  
  border: none;  
  padding: 15px 30px;  
  text-align: center;  
  font-size: 16px;  
  cursor: pointer;  
}  
  
.button-report {  
  font-size: 18px;  
}
```

CSS



## Conditional Statements

Koşullara bağlı olarak stil oluşturmak için kullanılırlar

```
@mixin avatar($size, $circle: false) {  
  width: $size;  
  height: $size;  
  
  @if $circle {  
    border-radius: $size / 2;  
  }  
}  
  
.square-av {  
  @include avatar(100px, false);  
}  
.circle-av {  
  @include avatar(100px, true);  
}
```

SCSS

```
.square-av {  
  width: 100px;  
  height: 100px;  
}  
  
.circle-av {  
  width: 100px;  
  height: 100px;  
  border-radius: 50px;  
}
```

CSS

```
@if $direction == up {  
  border-bottom-color: $color;  
}  
@else if $direction == right {  
  border-left-color: $color;  
}  
@else if $direction == down {  
  border-top-color: $color;  
}  
@else if $direction == left {  
  border-right-color: $color;  
}  
@else {  
  @error "Unknown direction #{ $direction }";  
}
```



## Loops

Döngü oluşturarak, belli değişimlere uygun olarak stil oluşturmak için kullanılır. For, while ve each döngüleri kullanılmaktadır.

```
$base-color: #036;  
  
@for $i from 1 through 3 {  
  ul:nth-child(3n + #{$i}) {  
    background-color:  
      lighten($base-color, $i * 5%);  
  }  
}
```

SCSS

```
ul:nth-child(3n + 1) {  
  background-color: #004080;  
}  
  
ul:nth-child(3n + 2) {  
  background-color: #004d99;  
}  
  
ul:nth-child(3n + 3) {  
  background-color: #0059b3;  
}
```

CSS