
Basic
Sass


Sass
Comments
/**/, //

주석(Comments) //, /**/

CSS 멀티라인 주석 뿐만 아니라, **JS 싱글라인 주석** 역시 **SASS에서 사용**할 수 있습니다.

yamoo9_example.html

yamoo9_example.html

```
/*
 * 여러 줄 주석의 경우는
 * CSS 컴파일 되어도 주석이 남아 있습니다.
 */
#multi-line-comments { content: "유지"; }


// 한 줄 주석의 경우는 CSS에서 지원하지 않기 때문에
// CSS 컴파일 시, 주석이 표시되지 않습니다.
#single-line-comments { content: "제거"; }
```

CSS 코드 변경시, 출력되지 않아요.

주석(Comments) //, /**/

*.sass 코드에서는 **CSS 주석의 시작부분에서 Tab 한 후, 주석을 입력합니다.**

```
// http://sass-
lang.com/documentation/file.SASS_REFERENCE.htm
l#comments

// SASS 한 줄 주석
// CSS 코드로 변경될 때, 해석되지 않습니다.

/*
 * CSS
 * 여러줄
 * 주석
 * CSS 코드로 변경될 때, 해석됩니다.

body
  font: 13px/1.5 "Nanum Gothic", Dotum, Sans-
Serief

/*
 * Nested Rule: 중첩 규칙
```

SASS

```
// http://sass-
lang.com/documentation/file.SASS_REFERENCE.htm
l#comments

// SASS 한 줄 주석
// CSS 코드로 변경될 때, 해석되지 않습니다.

/*
 * CSS
 * 여러줄
 * 주석
 * CSS 코드로 변경될 때, 해석됩니다.
 */

body {
    font: 13px/1.5 "Nanum Gothic", Dotum,
Sans-Serief;
}

/*
 * Nested Rule: 중첩 규칙
 */
```

SCSS

주석(Comments) 스니펫 만들기

Sublime Text에서 활용 가능한 주석 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
/*
 * .scss, .sass, .css 파일에서 활용
 * 여러 줄 주석 스니펫
 */
```

```
<snippet>
    <content><![CDATA[/*
 * $1
 */
$2]]></content>
    <tabTrigger>cmt</tabTrigger>
    <scope>source.sass, source.scss, source.css</scope>
    <description>CSS 여러줄 주석</description>
</snippet>
```

comments.sass

```
// http://sass-lang.com/documentation/file.
SASS_REFERENCE.html#comments

// SASS 한 줄 주석
// CSS 코드로 변경될 때, 해석되지 않습니다.

/*
 * CSS, SASS 여러 줄 주석
 * CSS 코드로 변경될 때, 해석됩니다.
 */
```

comments.css

```
/* CSS, SASS 여러 줄 주석
 * CSS 코드로 변경될 때, 해석됩니다.
 */
```

Nesting
Parent Selector

&

중첩 규칙(Nested Rules)

부모 선택자 **내부에** 자식 선택자를 **포함하는 구조 형태로 작성**이 가능합니다.

yamoo9_example.html

yamoo9_example.html

```
#page {                                  ●·····▶ #page
    content: "페이지";

    .container {                         ●·····▶ #page .container
        content: "콘테이너";

        .section {                       ●·····▶ #page .container .section
            content: "섹션"; }

    } // .container

} // #page
```

중첩 규칙(Nested Rules)

CSS 선택자 코드 작성은 효율성이 떨어지는 반면 **SASS 중첩규칙**은 매우 **효율성이 높습니다**.

yamoo9_example.html

yamoo9_example.html

```
1
```

```css
nav {float: right;}
nav li {float: left;}
nav li a {color: #666;}
nav li a:hover {color: #333;}
nav li.current {font-weight: bold;}
```

CSS

VS

```scss
nav {
  float: right;
  li {
    float: left;
    a {
      color: #666;
      &:hover {
        color: #333; }}
    &.current {
      font-weight: bold;
    }}}
```

yamoo9_example.html

yamoo9_example.html

```
div.container {
  div.content {
    div.articles {
      & > div.post {
        div.title {
          h1 {
            a {
            }
          }
        }
        div.content {
          ul {
            li { ... }
          }
        }
      }
    }
  }
}
```

Bad Nesting!

중첩 코드는 적당한게 좋아요.
너무 많은 중첩은 안 좋습니다. 왜냐하면...

SCSS

yamoo9_example.html

yamoo9_example.html

```
1

  div.content div.container { ... }
  div.content div.container div.articles { ... }
  div.content div.container div.articles > div.post { ... }
  div.content div.container div.articles > div.post div.title { ... }
  div.content div.container div.articles > div.post div.title h1 { ... }
  div.content div.container div.articles > div.post div.title h1 a { ... }
  div.content div.container div.articles > div.post div.content { ... }
  div.content div.container div.articles > div.post div.content ul { ... }
  div.content div.container div.articles > div.post div.content ul li { ... }
```

Oops!

적절하지 않은 중첩 코드는
쓸데없는 CSS 선택자를 남용하여 생성하기 때문입니다.

CSS

부모, 참조 선택자 (&)

앤퍼센트(&) 심볼은 **부모를 참조하는 선택자 역할**을 수행합니다.

yamoo9_example.html

```
p {
  font-weight: bold;
  text-decoration: none;

  &:hover a { text-decoration: underline; }
  #page.firefox &.showcase a { font-weight: normal; }
}
```

`#page.firefox p.showcase a`

`p:hover a`

yamoo9_example.html

yamoo9_example.html

```
a
    position: relative
    color: #495054

    &:hover
        color: #5f666d

    &.external-link,
    &[target=_blank]:before
        content: ''
        background: url(external_link.gif) no-repeat
        position: absolute
        right: 100%
        width: 10px
        height: 10px

    .article & .bold
        font-weight: bold
```

& 뒤에 '공간'이 있을 때와 없을 때,
& 앞에 선택자가 있을 경우 유의하여 보세요!

SASS

yamoo9_example.html

yamoo9_example.html

```scss
a {
  position: relative;
  color: #495054;
  &:hover {
    color: #5f666d;
  }
  &.external-link,
  &[target=_blank]:before {
    content: "";
    background: url(external_link.gif) no-repeat;
    position: absolute;
    right: 100%;
    width: 10px;
    height: 10px;
  }
  .article & .bold {
    font-weight: bold;
  }
}
```

& 뒤에 '공간'이 있을 때와 없을 때,
& 앞에 선택자가 있을 경우 유의하여 보세요!

SCSS

yamoo9_example.html

```
a {
  position: relative;
  color: #495054;
}
a:hover {
  color: #5f666d;
}
a.external-link, a[target=_blank]:before {
  content: "";
  background: url(external_link.gif) no-repeat;
  position: absolute;
  right: 100%;
  width: 10px;
  height: 10px;
}
.article a .bold {
  font-weight: bold;
}
```

& 뒤에 '공간'이 있을 때와 없을 때,
& 앞에 선택자가 있을 경우 유의하여 보세요!

CSS

중첩 부모 참조 선택자(&) 스니펫 만들기

Sublime Text에서 활용 가능한 중첩 부모 참조 선택자 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
<snippet>
    <content><![CDATA[&$1 {
    $2
}
$0]]></content>
    <tabTrigger>nested-parent</tabTrigger>
    <scope>source.scss, source.sass</scope>
    <description>SASS - 중첩 규칙, 부모 참조</description>
</snippet>
```

중첩 부모 참조 선택자(&) 스니펫 만들기

Sublime Text에서 활용 가능한 중첩 부모 참조 선택자 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
<snippet>
    <content><![CDATA[${1:상위 참조 선택자} &$2 {
    $3
}
$0]]></content>
    <tabTrigger>nested-parent-high</tabTrigger>
    <scope>source.scss, source.sass</scope>
    <description>SASS - 중첩 규칙, 상위 참조 선택자</description>
</snippet>
```

Combining Selectors

yamoo9_example.html

yamoo9_example.html

```
div {
  color: black

  // 선택 연산자
  .foo {
    color: black } // 자손(descendant) 선택자
  > .foo {
    color: black } // 자식(child) combinator
  + .foo {
    color: black } // 인접형제(adjacent sibling) 선택자
  ~ .foo {
    color: black } // 일반형제(general sibling) 선택자
  & .foo {
    color: black } // SASS 부모(Parent) 참조 선택자
  &.bar {
    color: black }
  &:hover {
    color: black }
}
```

중첩 속성(Nested Properties)

세분화된 속성 여러 개를 설정할 경우, **네임스페이스(Namespace)를 사용**하여 **중첩 속성을 설정**할 수 있습니다.

yamoo9_example.html

yamoo9_example.html

```
.yamoo9 {
  padding: {
    top: 10px;
    bottom: 5px;
  }
  font: {
    family: fantasy;
    size: 30em;
    weight: bold;
  }
}
```

'내부공간' 네임스페이스

'글꼴' 네임스페이스

CSS 속기형 속성
font, margin, border, padding, background, transition, animation, box, column, list, min, max, outline, text ….

중첩 속성 스니펫 만들기

Sublime Text에서 활용 가능한 중첩 속성 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
<snippet>
    <content><![CDATA[${1:margin,
animation, box, border, background, column, font, list, max, min,
outline, padding, text, transition}: {
    $2
}
$0]]></content>
    <tabTrigger>nested-properties</tabTrigger>
    <scope>source.scss, source.sass</scope>
    <description>SASS - 중첩 속성 정의</description>
</snippet>
```

RWD, 부모 참조 선택자와 미디어쿼리

선언 구간 내부의 미디어쿼리는 외부로 분리됩니다.

yamoo9_example.html

yamoo9_example.html

```
aside {
  width: 25%;
  @media screen and (max-width: 480px) {
    width: 100%;
  }
}
```

CSS 미디어쿼리

```
aside {
  width: 25%;
}

@media screen and (max-width: 480px) {
  aside {
    width: 100%;
  }
}
```

nesting-&.sass

```
/*
 * Nesting
 * 중첩 규칙을 사용하면 구조를 쉽게 파악하면서
 * 반복적인 선택자 코드 사용량을 대폭 줄일 수 있어
 * 매우 유용합니다.
 */

a
  position: relative
  color: #495054
  padding:
    top: 20px
    bottom: 10px

  @media only screen and (max-width: 480px)
    color: red

  &:hover
    color: #5f666d

  &.external-link,
  &[target=_blank]:before
    content: ''
    background: url(external_link.gif) no-repeat
    position: absolute
    right: 100%
    width: 10px
    height: 10px

    @media only screen and (max-width: 480px)
      position: static

  & .bold
    font-weight: bold
```

nesting-&.css

```
/* Nesting
 * 중첩 규칙을 사용하면 구조를 쉽게 파악하면서
 * 반복적인 선택자 코드 사용량을 대폭 줄일 수 있어
 * 매우 유용합니다.
 */
a {
  position: relative;
  color: #495054;
  padding-top: 20px;
  padding-bottom: 10px;
}
@media only screen and (max-width: 480px) {
  a {
    color: red;
  }
}
a:hover {
  color: #5f666d;
}
a.external-link, a[target=_blank]:before {
  content: "";
  background: url(external_link.gif) no-repeat;
  position: absolute;
  right: 100%;
  width: 10px;
  height: 10px;
}
@media only screen and (max-width: 480px) {
  a.external-link, a[target=_blank]:before {
    position: static;
  }
}
a .bold {
  font-weight: bold;
}
```

Selector Inheritance

@extend, %

선택자 상속(Selector Inheritance)

@extend를 사용해 **선언된 다른 규칙의 내용을 상속**할 수 있습니다.

yamoo9_example.html

yamoo9_example.html

```scss
.button {
  background-color: $color-main;
  font-weight: bold;
  color: white;
  padding: 5px;
}


.button-checkout {
  @extend .button;
  background-color: darken($color-main, 20%);
}
```

재사용 가능한 .button 클래스

.button 클래스 상속

선택자 상속(Selector Inheritance)

@extend를 사용해 **선언된 다른 규칙의 내용을 상속**할 수 있습니다.

yamoo9_example.html

yamoo9_example.html

```
.button-checkout {
  @extend .button;
  background-color: darken($color-main, 20%);

  .msg & {
    background-color: darken($color-main, 30%);
  }
}
```

선택자 상속(Selector Inheritance)

@extend를 사용해 **선언된 다른 규칙의 내용을 상속**할 수 있습니다.

yamoo9_example.html

yamoo9_example.html

```
1

.button, .button-checkout, .msg .button-checkout {
  background-color: blue;
  font-weight: bold;
  color: white;
  padding: 5px; }

.button-checkout {
  background-color: #000099; }

.msg .button-checkout {
  background-color: #000066; }
```

선택자 상속이란? 결과적으로 그룹핑이 된다는 점을 눈여겨 보세요!

CSS

inheritance.sass

```sass
/*
 * Selector Inheritance
 * 재사용 가능한 클래스 선택자를 상속할 수 있습니다.
 * CSS 코드 변경 시, 그룹핑으로 처리됩니다.
 */

// button 클래스
.button
  margin: 0 auto 25px
  padding: 15px
  border-radius: 6px
  em
    text-decoration: underline

// 실패 버튼
.button-fail
  @extend .button
  background-color: #fac9c7
  border-color: #ef625a

// 성공 버튼
.button-success
  @extend .button
  background-color: #1be4b7
  border-color: #bef8eb

// 비활성화 버튼
.button-disabled
  cursor: default
  @extend .button
  background-color: #c9c9c9
  border-color: #6c6c6c
  .msg &
    background-color: darken(#fac97c, 30%)
```

inheritance.css

```css
/* Selector Inheritance
 * 재사용 가능한 클래스 선택자를 상속할 수 있습니다.
 * CSS 코드 변경 시, 그룹핑으로 처리됩니다.
 */
.button, .button-fail, .button-success, .button-disabled {
  margin: 0 auto 25px;
  padding: 15px;
  border-radius: 6px;
}
.button em, .button-fail em, .button-success em, .button-disabled em {
  text-decoration: underline;
}

.button-fail {
  background-color: #fac9c7;
  border-color: #ef625a;
}

.button-success {
  background-color: #1be4b7;
  border-color: #bef8eb;
}

.button-disabled {
  cursor: default;
  background-color: #c9c9c9;
  border-color: #6c6c6c;
}
.msg .button-disabled {
  background-color: #d58508;
}
```

익스텐드(@extend) 스니펫 만들기

Sublime Text에서 활용 가능한 익스텐드 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
<snippet>
    <content><![CDATA[@extend ${1:상속할 선택자 이름}]]></content>
    <tabTrigger>/</tabTrigger>
    <scope>source.scss, source.sass</scope>
    <description>SASS - @extend 호출</description>
</snippet>
```

'/' 단축 키워드 말고 직접 'extend'로
등록하셔도 좋겠죠? ^ _ ^

선택자 상속, 옵션(!optional) 설정

@extend 오류 발생시, 오류가 발생되지 않도록 옵션 설정을 할 수 있습니다.

yamoo9_example.html

yamoo9_example.html

```scss
.button-checkout {
  @extend .button !optional;
  background-color: darken($color-main, 20%);

  .msg & {
    background-color: darken($color-main, 30%);
  }
}
```

@extend 상속 시, 선언된 선택자가 존재하지 않을 경우 '옵션' 처리하여 오류를 발생시키지 않습니다.

옵션(!optional) 플래그 스니펫 만들기

Sublime Text에서 활용 가능한 옵션 플래그 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
<snippet>
    <content><![CDATA[!optional]]></content>
    <tabTrigger>optional</tabTrigger>
    <description>SASS - !optional</description>
    <scope>source.scss, source.sass</scope>
</snippet>
```

대체 선택자(Placeholder Selector, %)

SASS는 **특수한 선택자 '%대체 선택자'를 지원**합니다. 해석된 **CSS에서는 출력 되지 않습니다.**

yamoo9_example.html

yamoo9_example.html

```
%reset-mp {
    margin: 0;
    padding: 0; }

#app {
    @extend %reset-mp; }

#design {
    @extend %reset-mp; }
```

% 선택자 코드는 CSS 코드 변경시, 출력되지 않아요.

placeholder.sass

```sass
/*
 * Placeholder % 선택자
 * CSS 코드 변경 시, 해석되지 않습니다.
 * HTML에 class 속성을 부여하지 않고도
 * 100% 시멘틱한 마크업이 가능합니다.
 */

%button
  margin: 0 auto 25px
  padding: 15px
  border-radius: 6px
  em
    text-decoration: underline

.button-fail
  @extend %button
  background-color: #fac9c7
  border-color: #ef625a

.button-success
  @extend %button
  background-color: #1be4b7
  border-color: #bef8eb

.button-disabled
  cursor: default
  @extend %button
  background-color: #c9c9c9
  border-color: #6c6c6c
  .msg &
    background-color: darken(#fac97c, 30%)

```

placeholder.css

```css
/* Placeholder % 선택자
 * CSS 코드 변경 시, 해석되지 않습니다.
 * HTML에 class 속성을 부여하지 않고도
 * 100% 시멘틱한 마크업이 가능합니다.
 */
.button-fail, .button-success, .button-disabled {
  margin: 0 auto 25px;
  padding: 15px;
  border-radius: 6px;
}
.button-fail em, .button-success em, .button-disabled em {
  text-decoration: underline;
}

.button-fail {
  background-color: #fac9c7;
  border-color: #ef625a;
}

.button-success {
  background-color: #1be4b7;
  border-color: #bef8eb;
}

.button-disabled {
  cursor: default;
  background-color: #c9c9c9;
  border-color: #6c6c6c;
}
.msg .button-disabled {
  background-color: #d58508;
}
```

플레이스홀더(%) 스니펫 만들기

Sublime Text에서 활용 가능한 플레이스홀더 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
<snippet>
    <content><![CDATA[%${1:플레이스홀더 이름} {
    $2
}
$0]]></content>
    <tabTrigger>,</tabTrigger>
    <scope>source.scss, source.sass</scope>
    <description>SASS - %placeholder 정의</description>
</snippet>
```

',' 단축 키워드 말고 직접 'placeholder'로 등록하셔도 좋겠죠? ^_^

OOCSS + Sass = The best way to CSS

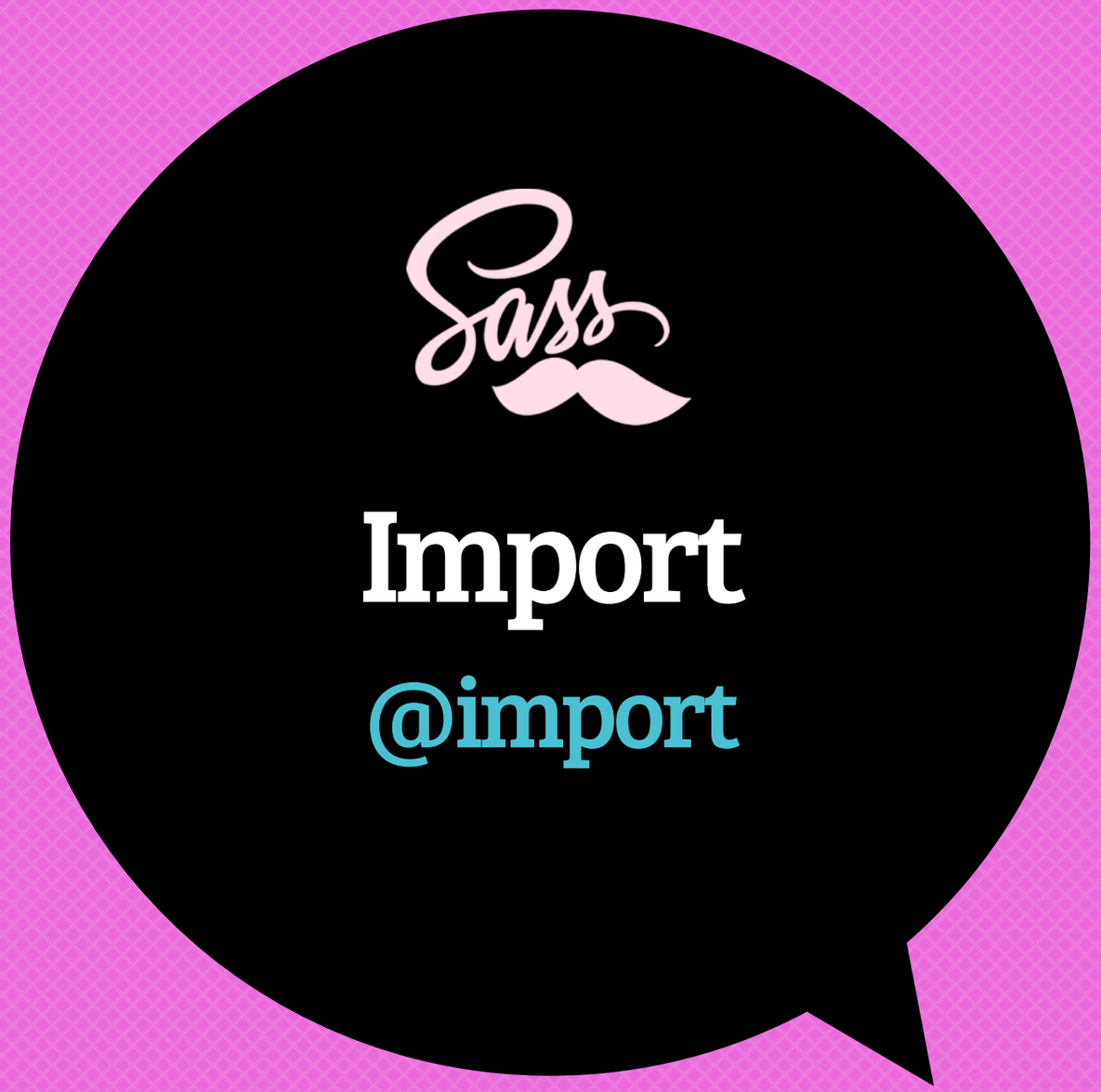
October 2012

Object-oriented CSS is awesome. But littering your markup with non-semantic classes is *not* awesome. Those classes sprinkled all over your HTML *are* going to change, and that's not gonna be fun. But if you combine OOCSS and Sass you get the best of both worlds: modular CSS without bloated, hard-to-maintain HTML.

IAN STORM TAYLOR
Co-founder @ Segment.io

모듈러 CSS

의미적이지 않은
표현 클래스 속성 문제


Sass
Import
@import

호출(Import on Steroids)

CSS 뿐만 아니라 ***.scss, *.sass 파일**을 **@import 문을 사용**하여 **호출** 할 수 있습니다.

yamoo9_example.html

yamoo9_example.html

```
1
// scss, sass 확장자의 경우는 생략할 수 있습니다.
@import "team-works.scss";
@import "team-works";

// 여러 개의 파일을 불러들일 때는 콤마(,)로 구분하여 불러들일 수 있습니다.
@import "team-works", "team-test";
```

CSS 확장자는 필히 명시해야 합니다.

import.sass

```sass
/*
 * @import
 * SASS 코드 상에서는 콤마(,)를 이용하여 구분자를 추가하면
 * 자동으로 @import 구문을 생성해줍니다.
 * 주석이 @import 문보다 먼저 위치한 경우에도
 * CSS 해석시에는 주석이 아래 존재하게 됩니다.
 */

@import "reset.css", "typography.css", "layout.css", "ie.css", "print.css"

/*
 * SASS 파일을 호출할 경우에는
 * 확장자를 생략해도 됩니다.
 */

@import "placeholder"
```

import.css

```css
@import url(reset.css);
@import url(typography.css);
@import url(layout.css);
@import url(ie.css);
@import url(print.css);
/* @import
 * SASS 코드 상에서는 콤마(,)를 이용하여 구분자를 추가하면
 * 자동으로 @import 구문을 생성해줍니다.
 * 주석이 @import 문보다 먼저 위치한 경우에도
 * CSS 해석시에는 주석이 아래 존재하게 됩니다.
 */
/* SASS 파일을 호출할 경우에는
 * 확장자를 생략해도 됩니다.
 */
/* Placeholder % 선택자
 * CSS 코드 변경 시, 해석되지 않습니다.
 * HTML에 class 속성을 부여하지 않고도
 * 100% 시멘틱한 마크업이 가능합니다.
 */
.button-fail, .button-success, .button-disabled {
  margin: 0 auto 25px;
  padding: 15px;
  border-radius: 6px;
}
.button-fail em, .button-success em, .button-disabled em {
  text-decoration: underline;
}

.button-fail {
  background-color: #fac9c7;
  border-color: #ef625a;
}

.button-success {
  background-color: #1be4b7;
  border-color: #bef8eb;
}

.button-disabled {
  cursor: default;
```

임포트(@import) 스니펫 만들기

Sublime Text에서 활용 가능한 임포트 스니펫을 만들어 볼까요?,

yamoo9_example.html

yamoo9_example.html

```
<snippet>
    <content><![CDATA[@import "$0";]]></content>
    <tabTrigger>import</tabTrigger>
    <scope>source.css, source.scss, source.sass</scope>
    <description>SASS - @import</description>
</snippet>
```

_호출 제어 (Partials)

*.scss, *.sass 파일 중 CSS 파일로 변경되지 않게 하는 방법은 파일 명 앞에 _을 붙여주면 됩니다.

yamoo9_example.html

yamoo9_example.html

```
// 기본적으로 SCSS, SASS 파일은 해석되어 CSS로 변경됩니다.
compile-to-css.sass

// SCSS, SASS 파일 앞에 밑줄(_)이 붙어 있으면 이를 CSS로 해석 변경하지 않습니다.
_decompile-to-css.sass
```

스타일 모듈 관리에 용이합니다.

스타일 모듈 관리 (Manage Style Modules)

SASS에서는 **각각의 역할을 분리**하여 **필요에 따라 재사용할 수 있도록 모듈 관리**하는 것이 가능합니다.

yamoo9_example.html

yamoo9_example.html

```
# style.sass
@import modules/normalize
@import modules/base
@import modules/mixins
@import partials/header
@import partials/footer
@import partials/ie
@import partials/print
```

style.sass

modules

- _normalize.sass
- _base.sass
- _mixins.sass

partials

- _header.sass
- _footer.sass
- ie.sass
- print.sass

스타일 모듈 관리 (Manage Style Modules)

SASS에서는 **각각의 역할을 분리**하여 **필요에 따라 재사용할 수 있도록 모듈 관리**하는 것이 가능합니다.

yamoo9_example.html

yamoo9_example.html

```
// 라이브러리: Libraries
@import "compass"
@import "public/config"
@import "public/reset"
@import "shared/avantgarde"
@import "public/mixins"
@import "public/sprites"
@import "shared/payment_types"

// 타이포그래피: Typography
@import "public/type_styles"
@import "public/type_layout"
@import "public/links"
@import "public/pagination"

// 폼: Forms
@import "public/form_styles"
@import "public/form_layout"
@import "public/messaging"

// 레이아웃: Layout
@import "public/layout"
@import "public/lightbox"
@import "public/images"
```

GROUP 1

import-modules.sass

GROUP 2

import-modules.css

FOLDERS

- sass-compass-example
  - .sass-cache
  - css
  - modules
    - _base.sass
    - _mixins.sass
    - _normalize.sass
  - partials
    - _footer.sass
    - _header.sass
    - _ie.sass
    - _print.sass
  - sass
  - cmt.sublime-snippet
  - compass_app_log.txt
  - config.rb
  - index.html

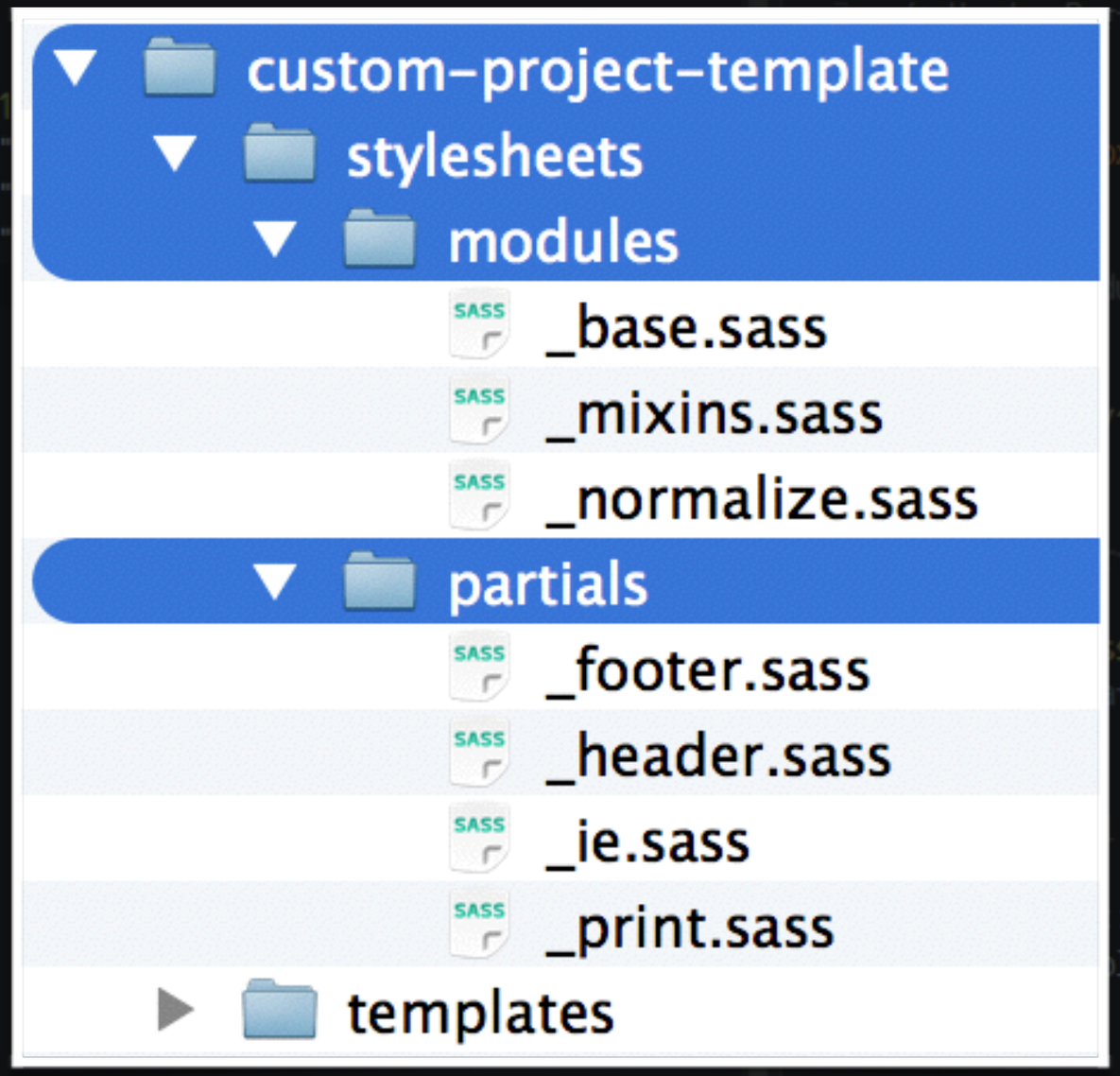
import-modules.sass

```sass
@import "../modules/normalize"
@import "../modules/base"
@import "../modules/mixins"
@import "../partials/header"
/*
 * import-modules
 */
body
  font: 13px/1.5 'Nanum Gothic', Dotum, Sans-Serief
@import "../partials/footer"
@import "../partials/ie"
@import "../partials/print"

```

import-modules.css

```css
/* Normalize Module
 */
/* Base Module
 */
/* Mixins Module
 */
/* Header Partial Module
 */
header {
  margin: 10px auto 15px;
}

/* import-modules
 */
body {
  font: 13px/1.5 "Nanum Gothic", Dotum, Sans-Serief;
}

/* Footer Partial Module
 */
footer address a {
  text-decoration: none;
}

/* IE Partial Module
 */
.lt-ie9 body::before {
  content: "old IE";
}

/* Print Partial Module
 */
@media print {
  body {
    color: #333333 !important;
    background-color: white;
  }
}
```

~/사용자계정/.compass



import-project-modules.sass

```sass
@import "modules/base"

@import "partials/header"

@import "partials/print"

@import "modules/normalize"
```

import-project-modules.css

```css
/* Base Module
 */
/* Header Partial Module
 */
header {
  margin: 10px auto 15px;
}

/* Print Partial Module
 */
@media print {
  body {
    color: #333333 !important;
    background-color: white;
  }
}
/* Normalize Module
 */
```

compass-reset.sass

```
// Compass reset 모듈 호출
@import "compass/reset"
```

compass-reset.css

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font: inherit;
  font-size: 100%;
  vertical-align: baseline;
}

html {
  line-height: 1;
}

ol, ul {
  list-style: none;
}

table {
```

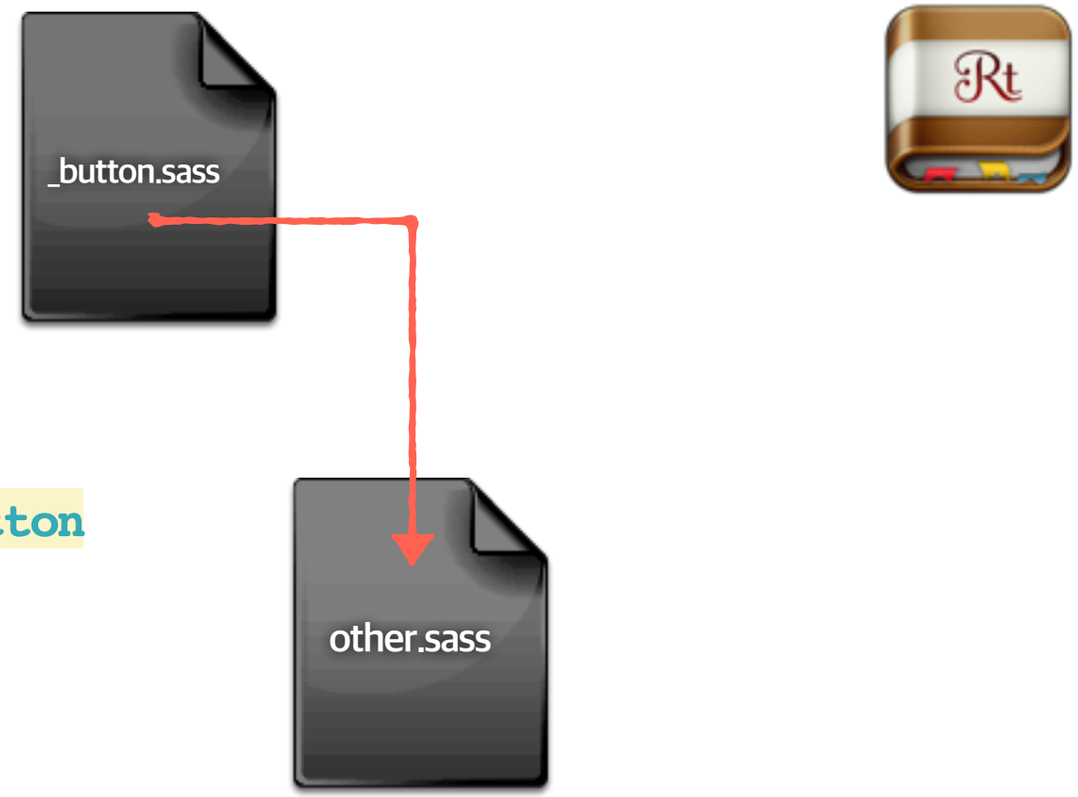
중첩 @import

@import를 선언 구문 내부에 사용하면 **별도로 분리된 파일에서 코드를 불러와 선택자 뒤에 불러온 모듈 선택자가 추가**됩니다. (선택자 상속)

yamoo9_example.html

```
.button {
  background-color: $color-main;
  font-weight: bold;
  color: white;
  padding: 5px;
}


#footer {
  @import button;
}
```

#footer .button



_box.sass

```sass
.box
  margin: 10px auto 20px
  padding: 10px
```

nested-import.sass

```sass
.demo
  @import box
  font: 1em Arial,sans-serif
```

nested-import.css

```css
.demo {
  font: 1em Arial, sans-serif;
}
.demo .box {
  margin: 10px auto 20px;
  padding: 10px;
}
```