scss 서식

**학습목표**

scss 서식 사용법

**목차**

[1. scss 서식 2](#_Toc136863116)

[1) 프로젝트 생성 2](#_Toc136863117)

[2) scss 예제 3](#_Toc136863118)

[3) 변수, darken 함수, lighten 함수 4](#_Toc136863119)

[4) @mixin, @include 5](#_Toc136863120)

[5) @if 7](#_Toc136863121)

[6) 클래스 이름 중첩 #1 8](#_Toc136863122)

[7) 클래스 이름 중첩 #2 10](#_Toc136863123)

[8) 클래스 이름 중첩 #3 12](#_Toc136863124)

[9) @for 14](#_Toc136863125)

[2. 반응형 서식 16](#_Toc136863126)

[1) 학생 목록 테이블 구현 16](#_Toc136863127)

[2) 반응형 서식 20](#_Toc136863128)

# scss 서식

## 프로젝트 생성

|  |
| --- |
| npx create-react-app scss1 --template typescript |

### sass 설치

|  |
| --- |
| npm i -D sass |

-D 개발할 때만 필요한 패키지 설치

sass: scss의 다른 이름

프로젝트를 빌드할 때, \*.scss 파일이 \*.css 파일로 트랜스파일된다.

sass 트랜스파일러는 빌드할 때만 필요하고, 실행할 때는 필요없다.

따라서 -D 옵션으로 설치한다.

## scss 예제

### App.scss

|  |
| --- |
| .btn {    display: inline-block;    padding: 0.5em 1.5em;    border: 1px solid gray;    border-radius: 0.5em;    background: linear-gradient(#fff, #ddd);    color: black;    text-decoration: none;  }  .btn:active {    -ms-transform: translateY(2px);    -webkit-transform: translateY(2px);    transform: translateY(2px);    background: #ccc;  } |

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <a className="btn">버튼1</a>      </div>    );  }  export default App |

실행

### App.scss 수정

|  |
| --- |
| .btn {    display: inline-block;    padding: 0.5em 1.5em;    border: 1px solid gray;    border-radius: 0.5em;    background: linear-gradient(#fff, #ddd);    color: black;    text-decoration: none;    &:active {      -ms-transform: translateY(2px);      -webkit-transform: translateY(2px);      transform: translateY(2px);      background: #ccc;    }  } |

& -> 현재 태그를 의미함.

즉 &:active -> .btn:active

## 변수, darken 함수, lighten 함수

### App.scss

|  |
| --- |
| $color: #88f;  $color1: darken($color, 15%);  $color2: lighten($color, 15%);  span { width: 100px; height: 100px; margin: 20px; display: inline-block; }  span.a { background-color: $color1; }  span.b { background-color: $color; }  span.c { background-color: $color2; } |

$color: #88f;

변수 이름이 $color

변수에 대입된 값이 #88f

darken(색, %) lighten(색, %) 기본 함수

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <span className="a"></span>        <span className="b"></span>        <span className="c"></span>      </div>    );  }  export default App |

A blue square with white border

Description automatically generated with low confidence

### App.scss 수정

|  |
| --- |
| $color: #88f;  $color1: darken($color, 15%);  $color2: lighten($color, 15%);  span {    width: 100px; height: 100px; margin: 20px; display: inline-block;    &.a { background-color: $color1; }    &.b { background-color: $color; }    &.c { background-color: $color2; }  } |

## @mixin, @include

### App.scss

|  |
| --- |
| div.blue {    $color: #88f;    $color1: darken($color, 15%);    $color2: lighten($color, 15%);    span {      width: 100px; height: 100px; margin: 20px; display: inline-block;      &.a { background-color: $color1; }      &.b { background-color: $color; }      &.c { background-color: $color2; }    }  }  div.red {    $color: #f88;    $color1: darken($color, 15%);    $color2: lighten($color, 15%);    span {      width: 100px; height: 100px; margin: 20px; display: inline-block;      &.a { background-color: $color1; }      &.b { background-color: $color; }      &.c { background-color: $color2; }    }  } |

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <div className="blue">          <span className="a"></span>          <span className="b"></span>          <span className="c"></span>        </div>        <div className="red">          <span className="a"></span>          <span className="b"></span>          <span className="c"></span>        </div>      </div>    );  }  export default App |

A picture containing rectangle, colorfulness, lilac, square

Description automatically generated

### App.scss 수정

|  |
| --- |
| @mixin my-span($color) {    $color1: darken($color, 15%);    $color2: lighten($color, 15%);    span {      width: 100px; height: 100px; margin: 20px; display: inline-block;      &.a { background-color: $color1; }      &.b { background-color: $color; }      &.c { background-color: $color2; }    }  }  div.blue {    @include my-span(#88f);  }  div.red {    @include my-span(#f88);  } |

### App.tsx 수정

|  |
| --- |
| import './App.scss'  function colorDiv(name: string) {    return (      <div className={name}>        <span className="a"></span>        <span className="b"></span>        <span className="c"></span>      </div>    )  }  function App() {    return (      <div>        {colorDiv("blue")}        {colorDiv("red")}      </div>    );  }  export default App |

## @if

### App.scss

|  |
| --- |
| @mixin my-btn($color) {    $color\_dark: darken($color, 10%);    $color\_light: lighten($color, 10%);    $font\_color: white;    @if (lightness($color) >= 80) {      $font\_color: black;    }    display: inline-block;    padding: 0.5em 1.5em;    border: 1px solid $color\_dark;    border-radius: 0.4em;    background: linear-gradient($color\_light, $color\_dark);    color: $font\_color;    text-decoration: none;    &:active {      -ms-transform: translateY(0.1em);      -webkit-transform: translateY(0.1em);      transform: translateY(0.1em);      background: $color\_dark;    }  }  .my-gray-btn {    @include my-btn(#ddd);  }  .my-blue-btn {    @include my-btn(#55f);  }  .my-red-btn {    @include my-btn(#f55);  }  a { margin: 10px } |

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <a href="#" className="my-gray-btn">버튼1</a>        <a href="#" className="my-blue-btn">버튼2</a>        <a href="#" className="my-red-btn">버튼3</a>      </div>    );  }  export default App |

A blue rectangle with white text

Description automatically generated

## 클래스 이름 중첩 #1

### App.scss

|  |
| --- |
| @mixin my-btn($color) {    $color\_dark: darken($color, 10%);    $color\_light: lighten($color, 10%);    $font\_color: white;    @if (lightness($color) >= 80) {      $font\_color: black;    }    display: inline-block;    padding: 0.5em 1.5em;    border: 1px solid $color\_dark;    border-radius: 0.4em;    background: linear-gradient($color\_light, $color\_dark);    color: $font\_color;    text-decoration: none;    &:active {      -ms-transform: translateY(0.1em);      -webkit-transform: translateY(0.1em);      transform: translateY(0.1em);      background: $color\_dark;    }  }  .btn {    &-gray {      @include my-btn(#ddd);    }    &-blue {      @include my-btn(#55f);    }    &-red {      @include my-btn(#f55);    }  }  a { margin: 10px } |

  .btn-gray {

    @include my-btn(#ddd);

  }

  .btn-blue {

    @include my-btn(#55f);

  }

  .btn-red {

    @include my-btn(#f55);

  }

& 부분이 현재 태그인 .btn으로 위와 같이 치환된다.

css 문법에서 - 문자는 특별한 의미가 없고 알파벳 처럼 이름의 일부 문자이다.

btn-gray, btn-blue, btn-red 세 개의 이름.

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <a href="#" className="btn-gray">버튼1</a>        <a href="#" className="btn-blue">버튼2</a>        <a href="#" className="btn-red">버튼3</a>      </div>    );  }  export default App |

## 클래스 이름 중첩 #2

### App.scss

|  |
| --- |
| @mixin my-btn($color) {    $color\_dark: darken($color, 10%);    $color\_light: lighten($color, 10%);    $font\_color: white;    @if (lightness($color) >= 80) {      $font\_color: black;    }    display: inline-block;    padding: 0.5em 1.5em;    border: 1px solid $color\_dark;    border-radius: 0.4em;    background: linear-gradient($color\_light, $color\_dark);    color: $font\_color;    text-decoration: none;    &:active {      -ms-transform: translateY(0.1em);      -webkit-transform: translateY(0.1em);      transform: translateY(0.1em);      background: $color\_dark;    }  }  .btn {    margin: 10px;    &.gray {      @include my-btn(#ddd);    }    &.blue {      @include my-btn(#55f);    }    &.red {      @include my-btn(#f55);    }  } |

  .btn.gray {

    @include my-btn(#ddd);

  }

  .btn.blue {

    @include my-btn(#55f);

  }

  .btn.red {

    @include my-btn(#f55);

  }

css 실렉터에서 . 문자는 이름의 일부가 아니고 class 이름을 지정하는 특수 문자이다.

그래서 .btn.gray 이것은 class="btn gray" 를 의미한다.

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <a href="#" className="btn gray">버튼1</a>        <a href="#" className="btn blue">버튼2</a>        <a href="#" className="btn red">버튼3</a>      </div>    );  }  export default App |

## 클래스 이름 중첩 #3

### App.scss

|  |
| --- |
| @mixin my-btn($color) {    $color\_dark: darken($color, 10%);    $color\_light: lighten($color, 10%);    $font\_color: white;    @if (lightness($color) >= 80) {      $font\_color: black;    }    display: inline-block;    padding: 0.5em 1.5em;    border: 1px solid $color\_dark;    border-radius: 0.4em;    background: linear-gradient($color\_light, $color\_dark);    color: $font\_color;    text-decoration: none;    &:active {      -ms-transform: translateY(0.1em);      -webkit-transform: translateY(0.1em);      transform: translateY(0.1em);      background: $color\_dark;    }  }  .btn {    margin: 10px;    @include my-btn(#ddd);    &.blue {      @include my-btn(#55f);    }    &.red {      @include my-btn(#f55);    }    &.small {      font-size: 0.9em;    }    &.large {      font-size: 1.2em;    }  } |

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <a href="#" className="btn blue small">버튼1</a>        <a href="#" className="btn">버튼2</a>        <a href="#" className="btn red large">버튼3</a>      </div>    );  }  export default App |

A white rectangle with black and blue text

Description automatically generated

## @for

### App.tsx

|  |
| --- |
| import './App.scss'  function App() {    return (      <div>        <span>1</span>        <span>2</span>        <span>3</span>        <span>4</span>        <span>5</span>      </div>    );  }  export default App |

### App.scss

|  |
| --- |
| div {    span {      display: inline-block;      margin: 20px;      border: 1px solid gray;    }    span:nth-child(1) {      width: 50px;      height: 50px;      background-color: #888;    }    span:nth-child(2) {      width: 60px;      height: 60px;      background-color: #99f;    }    span:nth-child(3) {      width: 75px;      height: 75px;      background-color: #ffa;    }    span:nth-child(4) {      width: 90px;      height: 90px;      background-color: #afa;    }    span:nth-child(5) {      width: 120px;      height: 120px;      background-color: #fcc;    }  } |

A picture containing rectangle, post-it note, design

Description automatically generated

### App.scss

|  |
| --- |
| $width: 50px, 60px, 75px, 90px, 120px;  $color: #888, #99f, #ffa, #afa, #fcc;  div {    span {      display: inline-block;      margin: 20px;      border: 1px solid gray;    }    @for $i from 1 through length($width) {      span:nth-child(#{$i}) {        width: nth($width, $i);        height: nth($width, $i);        background-color: nth($color, $i);      }    }  } |

# 반응형 서식

## 학생 목록 테이블 구현

### StudentList.tsx

|  |
| --- |
| import './StudentList.scss'  const students = [    {id: 1, studentNo: '201014101', name: '홍길동', email: 'hong@skhu.net' },    {id: 2, studentNo: '201014102', name: '이몽룡', email: 'lee@skhu.net' },    {id: 3, studentNo: '201014103', name: '임꺽정', email: 'lim@skhu.net' },    {id: 4, studentNo: '201014104', name: '연흥부', email: 'yeong@skhu.net' },  ];  function StudentList() {    return (      <div id="StudentList">        <div className="table">          <div className="row">            <div>id</div>            <div>학번</div>            <div>이름</div>            <div>이메일</div>          </div>          { students.map(student => (              <div className="row" key={student.id}>                <div>{ student.id }</div>                <div>{ student.studentNo }</div>                <div>{ student.name }</div>                <div>{ student.email }</div>              </div>          ))}          </div>      </div>    )  }  export default StudentList |

### StudentList.scss

|  |
| --- |
| div#StudentList {    $border: 1px solid gray;    .table {      display: table; /\* <table> 태그처럼 보여진다 \*/      width: 100%;      border-left: $border;      border-top: $border;      .row { /\* .table 아래의 .row 태그에 서식 지정 \*/        display: table-row; /\* <tr> 태그처럼 보여진다 \*/        /\* .table 아래 .row:nth-child(1) 태그에 서직 시정 \*/        &:nth-child(1) { background-color: #eee; }        /\* .table 아래 .row 아래 div 태그에 서식 지정 \*/        & div {          display: table-cell; /\* <td> 태그처럼 보여진다 \*/          border-right: $border;          border-bottom: $border;          padding: 5px;        }      }    }  } |

### App.tsx

|  |
| --- |
| import './App.scss'  import StudentList from './StudentList'  function App() {    return (      <div id="App">        <StudentList />      </div>    );  }  export default App |

### App.scss

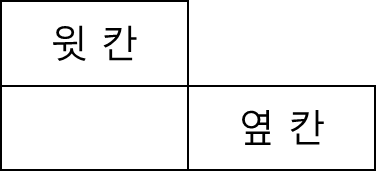
|  |
| --- |
| div#App {    width: 600px;    margin: 0 auto; padding: 20px;    border: 1px solid gray;    box-shadow: 3px 3px 3px #aaa;  } |

### 실행

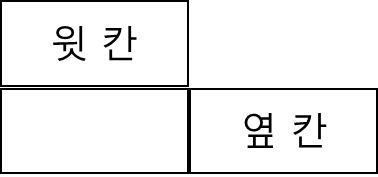
A picture containing text, screenshot, font, number

Description automatically generated

### td 태그의 border



### div 태그의 border



위와 같이 border가 곂치는 것을 막으려면,

각 칸은 border-right와 border-bottom만 그려야 한다.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

그 다음에 table의 border-left와 border-top을 추가로 그린다

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

## 반응형 서식

### App.scss

|  |
| --- |
| div#App {    width: 600px;    margin: 0 auto; padding: 20px;    border: 1px solid gray;    box-shadow: 3px 3px 3px #aaa;  }  /\* 위의 서식에 추가로, 창의 크기가 650px 이하일 때만 적용할 서식 \*/  @media (max-width: 650px) {    body { margin: 0; padding: 5px; }    div#App {      width: 100%;      margin: 0; padding: 0;      border: none;      box-shadow: none;    }  } |

A screenshot of a computer

Description automatically generated with medium confidence

창의 크기가 클 때

A screenshot of a computer

Description automatically generated

창 크기가 650px 이하일 때

### StudentList.scss

|  |
| --- |
| div#StudentList {    $border: 1px solid gray;    .table {      display: table;      width: 100%;      border-left: $border;      border-top: $border;      .row {        display: table-row;        &:nth-child(1) { background-color: #eee; }        div {          display: table-cell;          border-right: $border;          border-bottom: $border;          padding: 5px;        }      }    }  }  @media (max-width: 500px) {    div#StudentList {      .table {        display: block; width: 100%; border: none;        /\* .table 아래 .row 태그 서식 \*/        .row {          display: block; padding: 5px;          border: 1px solid gray;          box-shadow: 3px 3px 3px #ddd;          border-radius: 5px;          /\* .table 아래 .row 태그 바로 다음의 .row 태그에 서식 지정 \*/          & + .row { margin-top: 5px; }          /\* .table 아래 .row:nth-child(1) 태그에 서식 지정 \*/          &:nth-child(1) { display: none; }          /\* .table 아래 .row 아래 div 태그 서식 \*/          & div { display: block; border: none; padding: 5px; }          /\* .table 아래 .row 아래 div 태그 앞 부분에 문자열 삽입 \*/          & div:nth-child(1)::before { content: 'ID: '; }          & div:nth-child(2)::before { content: '학번: ' }          & div:nth-child(3)::before { content: '이름: ' }          & div:nth-child(4)::before { content: 'email: ' }          /\* 삽입된 문자열에 서식 지정 \*/          & div::before {            display: inline-block;            width: 50px;            text-align: right;            margin-right: 10px;            color: #888;            font-size: 0.9em;          }        }      }    }  } |

A screenshot of a phone

Description automatically generated with medium confidence

창 크기가 500px 이하일 때