

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

게시글 수정하기 - Modify

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
13 @Slf4j
14 @Controller
15 public class EighthController {
16
17     @Autowired
18     private BoardService service;
19
20     @GetMapping("/modify")
21     @RequestMapping("/modify")
22     public String getModify (int boardNo, Model model) throws Exception {
23         log.info("getModify()");
24
25         // 수정을 눌렀을때 기존 정보를 가져와야함
26         model.addAttribute(service.read(boardNo));
27
28         return "/board/eighth/modify";
29     }
30
31     @PostMapping("/modify")
32     @RequestMapping("/modify")
33     public String postModify (Board board, Model model) throws Exception {
34         log.info("postModify(): " + board);
35
36         service.modify(board);
37         model.addAttribute( attributeName: "msg", attributeValue: "수정이 성공적으로 완료되었습니다!");
38
39         return "/board/fourth/success";
40     }
41 }
```

Repository

```
107 //eighth - modify
108
109 @Repository
110 public void update(Board board) throws Exception {
111     String query = "update board set title = ?, content = ? where board_no = ?";
112
113     jdbcTemplate.update(query, board.getTitle(), board.getContent(), board.getBoardNo());
114 }
```

Service

```
19 // eighth
20 // modify
21 public void modify(Board board) throws Exception;
22
23 }
```

ServiceImpl

```
40 // eighth
41 // modify
42 @Override
43 public void modify(Board board) throws Exception {
44     repository.update(board);
45 }
46 }
```

게시글 수정하기 - Modify

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>

<script type="text/javascript" src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script>
  $(document).ready(function() {
    var formObj = $("#board");

    $("#btnModify").on("click", function() {
      formObj.attr("action", "modify");
      formObj.attr("method", "post");
      formObj.submit();
    });

    $("#btnList").on("click", function() {
      self.location = "lists";
    });
  });
</script>
```

```
<body>
<h2>게시글 수정</h2>

<form id="board" th:object="${board}" method="post">
  <input type="hidden" name="boardNo" th:field="*{boardNo}"/>

  <table>
    <tr>
      <td>제목</td>
      <td><input type="text" name="title" th:field="*{title}"/></td>
    </tr>
    <tr>
      <td>작성자</td>
      <td><input type="text" name="writer" th:field="*{writer}" readonly="readonly"/></td>
    </tr>
    <tr>
      <td>본문</td>
      <td><textarea cols="50" rows="20" name="content" th:field="*{content}"/></textarea></td>
    </tr>
  </table>
</form>

<div>
  <button type="submit" id="btnModify">게시글 수정</button>
  <button type="submit" id="btnList">게시글 목록 보기</button>
</div>
</body>
</html>
```

게시글 수정하기 - Modify

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

생성된 html

← → ↻ ⓘ localhost:7777/lists

게시판 목록 보기

[글쓰기](#)

번호	
15	글
14	변경테스트
13	tt
12	공지사항
11	공
10	게
8	1
6	ss
1	d



게시글 읽기

제목
작성자
기

본문

[게시글 수정](#) [게시글 삭제](#) [게시글 목록 보기](#)

게시글 수정

제목
작성자
기

본문

[게시글 수정](#) [게시글 목록 보기](#)



← → ↻ ⓘ localhost:7777/modify

동작 성공!

수정이 성공적으로 완료되었습니다!

[게시글 목록보기](#)



← → ↻ ⓘ localhost:7777/lists

게시판 목록 보기

[글쓰기](#)

번호	
15	글 수정
14	변경테스트

CSS – Basic

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

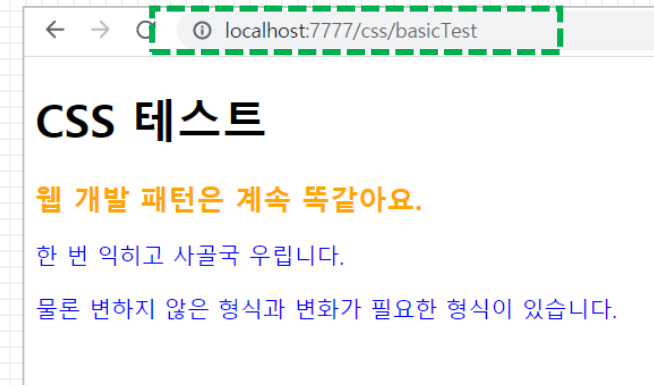
컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
14
15      // 결국 애는 /css/basicTest와 같음
16      @GetMapping("/basicTest")
17      public String getBasicCssTest() {
18          log.info("getBasicCssTest()");
19
20          return "csstest/eighth/cssTest";
21      }
22  }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <style>
7          h2 {
8              font-size: 20px;
9              color: orange;
10         }
11         p {
12             color: blue;
13         }
14     </style>
15 </head>
16 <body>
17     <h1>CSS 테스트</h1>
18     <h2>웹 개발 패턴은 계속 똑같아요.</h2>
19     <p>한 번 익히고 사골국 우립니다.</p>
20     <p>물론 변하지 않은 형식과 변화가 필요한 형식이 있습니다.</p>
21 </body>
22 </html>
```

생성된 html



CSS – Style

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
14
15
16
17
18
19
20
21
22
23  @GetMapping("/styleTest")
24  public String getStyleTest() {
25      log.info("getStyleTest()");
26
27      return "csstest/eighth/styleTest";
28  }
```

CSS

```
1  h2{
2      color: blue;
3  }
4  p{
5      font-size: 0.9em; /* 글자 크기 */
6      line-height: 2.0; /* 줄 간격 */
7      color: purple;
8  }
```

생성된 html

← → ↻ localhost:7777/css/styleTest

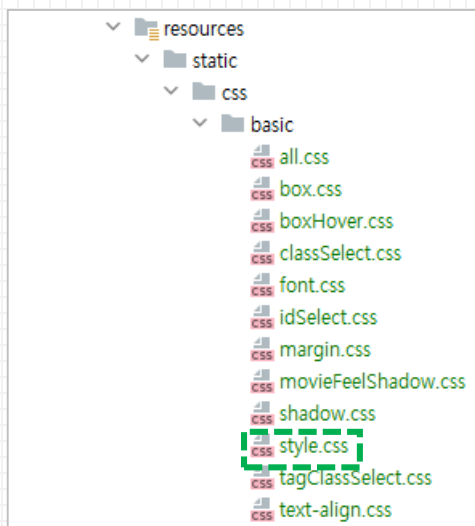
CSS 테스트

웹 개발 패턴은 계속 똑같아요.

한 번 익히고 사골국 우립니다.

물론 변하지 않은 형식과 변화가 필요한 형식이 있습니다.

CSS 위치



html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/style.css">
7  </head>
8  <body>
9      <h1>CSS 테스트</h1>
10     <h2>웹 개발 패턴은 계속 똑같아요.</h2>
11     <p>한 번 익히고 사골국 우립니다.</p>
12     <p>물론 변하지 않은 형식과 변화가 필요한 형식이 있습니다.</p>
13 </body>
14 </html>
```

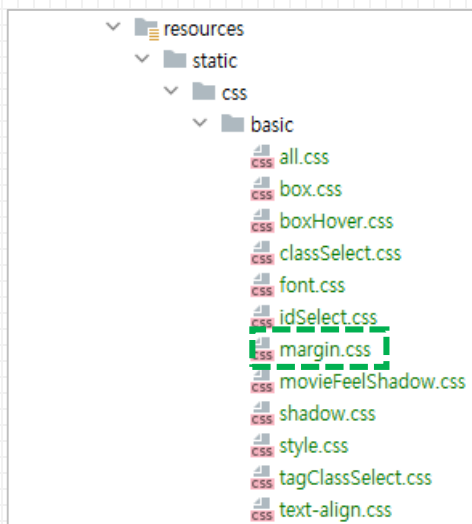
CSS – Margin

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30  @GetMapping("/marginTest")
31  public String getMarginTest() {
32      log.info("getMarginTest()");
33
34      return "csstest/eighth/marginTest";
35  }
```

CSS 위치



CSS

```
1  h2 {
2      color: blue;
3  }
4  p {
5      font-size: 12px;
6      margin-left: 20px;
7  }
```

생성된 html



html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/margin.css">
7  </head>
8  <body>
9      <h1>CSS 테스트</h1>
10     <h2>웹 개발 패턴은 계속 똑같아요.</h2>
11     <p>한 번 익히고 사골국 우립니다.</p>
12     <p>물론 변하지 않은 형식과 변화가 필요한 형식이 있습니다.</p>
13 </body>
14 </html>
```



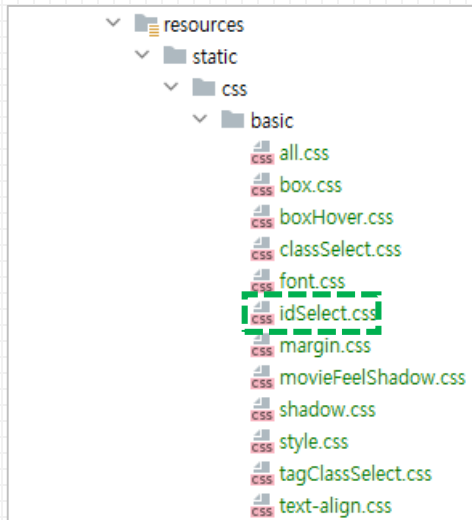
CSS – # (idSelect)

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37  @GetMapping("/idSelectTest")
38  public String getIdSelectTest() {
39      log.info("getIdSelectTest()");
40
41      return "csstest/eighth/idSelectTest";
42  }
```

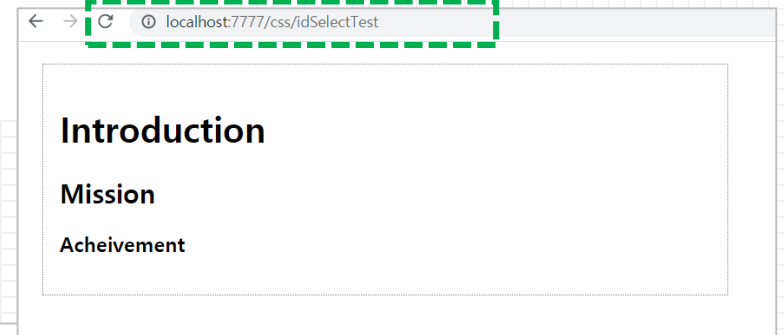
CSS 위치



CSS

```
1  #container {
2      width: 600px;           /* 너비 */
3      padding: 15px;         /* 테두리와 내용 사이의 여백 */
4      border: 1px dotted gray; /* 테두리 스타일 */
5      margin: 20px;
6  }
7
8  /* # 으로 하면 id로 선택하여 부를 수 있다. */
```

생성된 html



html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/idSelect.css">
7  </head>
8  <body>
9      <div id="container">
10         <h1>Introduction</h1>
11
12         <h2>Mission</h2>
13
14         <h3>Acheivement</h3>
15     </div>
16 </body>
17 </html>
```


CSS - * (all)

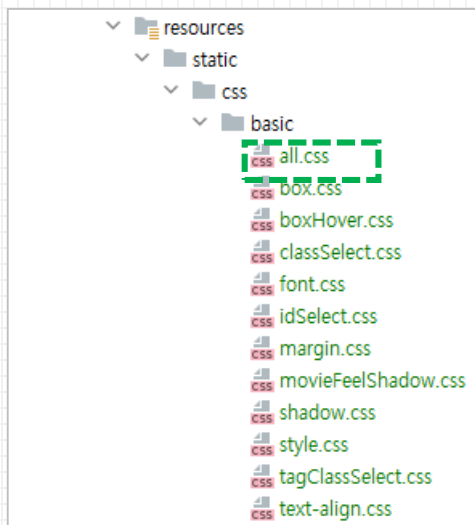
링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9 @Slf4j
10 @Controller
11 // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12 @RequestMapping(value = "/css")
13 public class EightCssController {
```

```
44 @GetMapping("/allTest")
45 public String getAllTest() {
46     log.info("getAllTest()");
47
48     return "csstest/eighth/allTest";
49 }
```

CSS 위치



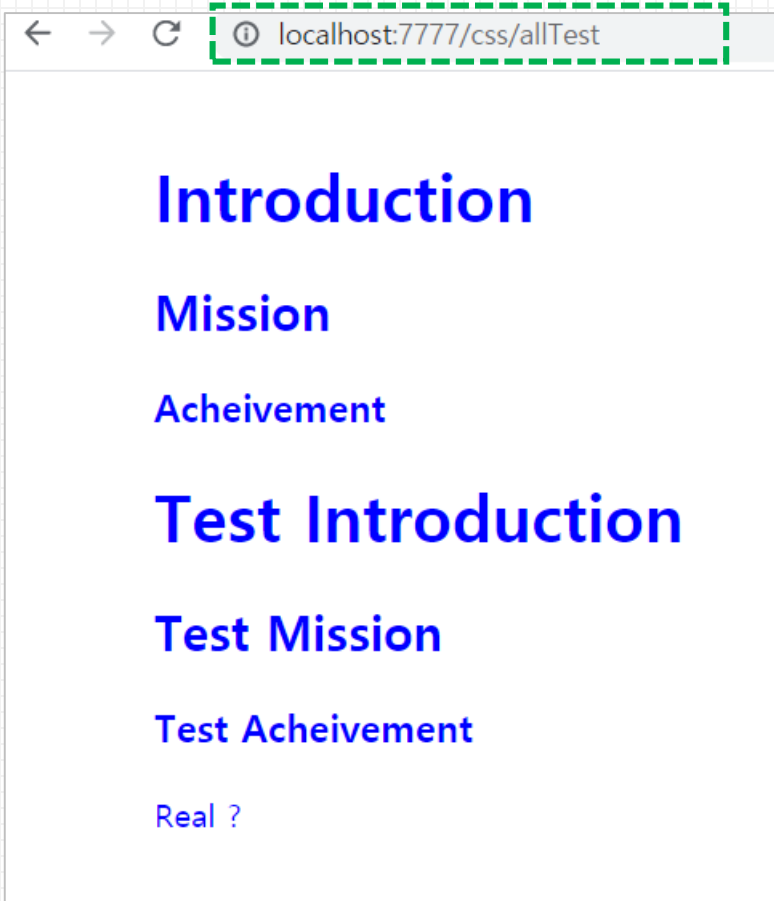
CSS

```
1 * {
2     margin: 20px;
3     color: blue;
4 }
5 /* 모든 정보에 적용 */
```

html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <link rel="stylesheet" href="/css/basic/all.css">
7 </head>
8 <body>
9     <div id="container">
10         <h1>Introduction</h1>
11
12         <h2>Mission</h2>
13
14         <h3>Acheivement</h3>
15     </div>
16     <div id="test">
17         <h1>Test Introduction</h1>
18
19         <h2>Test Mission</h2>
20
21         <h3>Test Acheivement</h3>
22     </div>
23     <div>
24         <p>Real ? </p>
25     </div>
26 </body>
27 </html>
```

생성된 html





CSS - . (class)

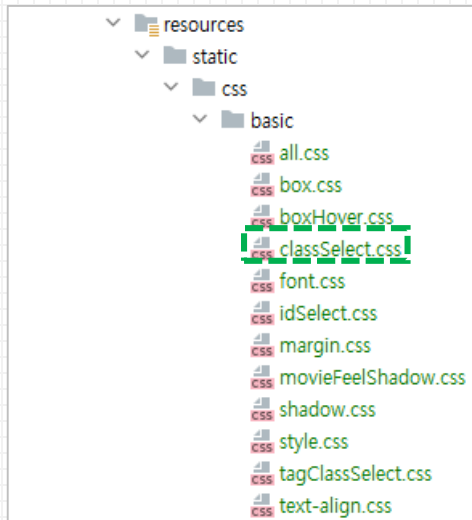
링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
```

```
51  @GetMapping("/classSelectTest")
52  public String getClassSelectTest() {
53      log.info("getClassSelectTest()");
54
55      return "csstest/eighth/classSelectTest";
56  }
```

CSS 위치



CSS

```
1  .greenBoldText {
2      color: green;
3      font-weight: bold;
4  }
5  /* . 으로 하면 class로 선택하여 부를 수 있다. */
```

생성된 html

← → ↻ ⓘ localhost:7777/css/classSelectTest

Introduction

Mission

Acheivement

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/classSelect.css">
7  </head>
8  <body>
9      <div id="container">
10         <h1>Introduction</h1>
11
12         <h2 class="greenBoldText">Mission</h2>
13
14         <h3>Acheivement</h3>
15     </div>
16 </body>
17 </html>
```

CSS – Tag class

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
```

```
58  @GetMapping("/tagClassSelectTest")
59  public String getTagClassSelectTest() {
60      log.info("getTagClassSelectTest()");
61
62      return "csstest/eighth/tagClassSelectTest";
63  }
```

CSS

```
1  h2.greenHighlight {
2      color: green;
3      font-weight: bold;
4      background-color: black;
5  }
```

생성된 html

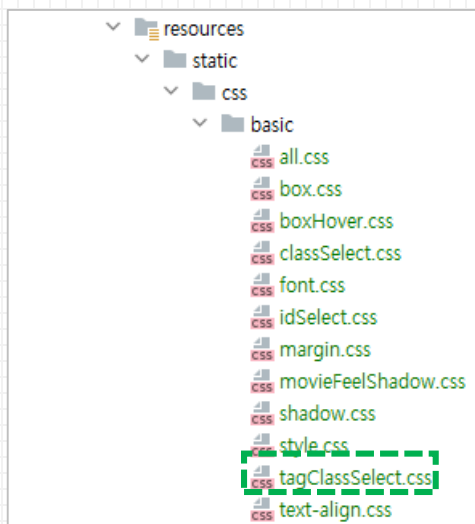
localhost:7777/css/tagClassSelectTest

Introduction

Mission

Acheivement

CSS 위치



html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/tagClassSelect.css">
7  </head>
8  <body>
9      <div id="container">
10         <h1>Introduction</h1>
11
12         <h2 class="greenHighlight">Mission</h2>
13
14         <h3>Acheivement</h3>
15     </div>
16 </body>
17 </html>
```



CSS – Box

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

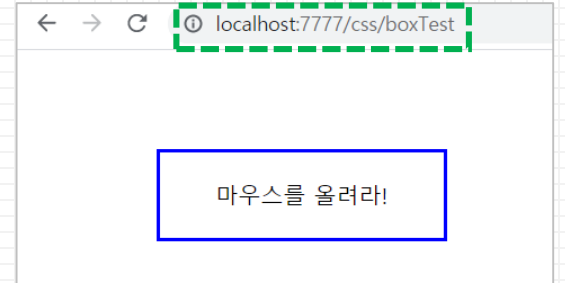
```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
```

```
65  @GetMapping("/boxTest")
66  public String getBoxTest() {
67      log.info("getBoxTest()");
68
69      return "csstest/eighth/boxTest";
70  }
```

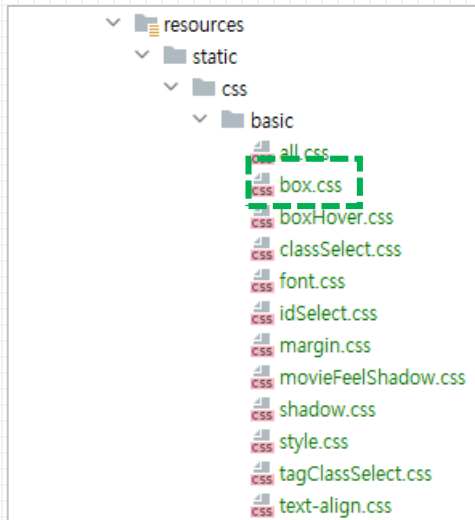
CSS

```
1  .box {
2      position: absolute; /* 박스가 배치되는 위치를 절대값으로 지정 */
3      left: 100px;
4      top: 70px;
5      width: 200px;
6      height: 60px;
7      background: #ffffff;
8      border: 3px solid blue;
9      text-align: center; /* 글자 위치 가운데 적용 */
10     line-height: 60px;
11 }
```

생성된 html



CSS 위치



html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/box.css">
7  </head>
8  <body>
9      <div class="box">마우스를 올려라!</div>
10 </body>
11 </html>
```

CSS – Box Hover

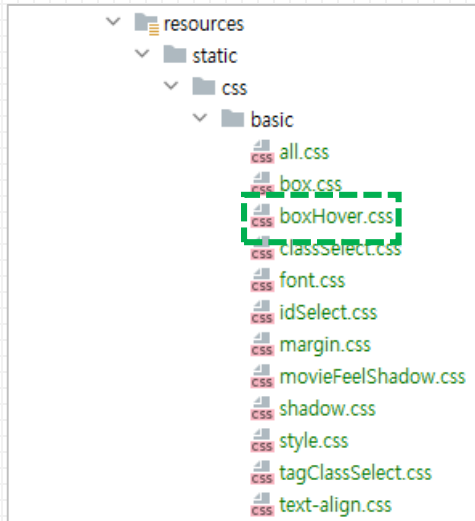
링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
```

```
72  @GetMapping("/boxHoverTest")
73  public String getBoxHoverTest() {
74      log.info("getBoxHoverTest()");
75
76      return "csstest/eighth/boxHoverTest";
77  }
```

CSS 위치



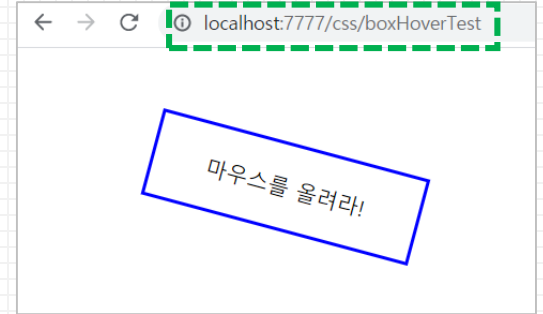
CSS

```
1  .box {
2      position: absolute; /* 박스가 배치되는 위치를 절대값으로 지정 */
3      left: 100px;
4      top: 70px;
5      width: 200px;
6      height: 60px;
7      background: #ffffff;
8      border: 3px solid blue;
9      text-align: center; /* 글자 위치 가운데 적용 */
10     line-height: 60px;
11 }
12 .box:hover {
13     -webkit-transform: rotate(15deg);
14     -moz-transform: rotate(15deg);
15     -o-transform: rotate(15deg);
16     -ms-transform: rotate(15deg);
17     transform: rotate(15deg);
18 }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/boxHover.css">
7  </head>
8  <body>
9      <div class="box">마우스를 올려라!</div>
10 </body>
11 </html>
```

생성된 html



CSS – Font

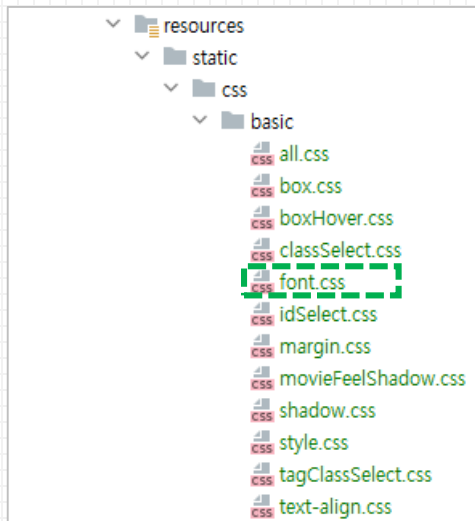
링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9 @Slf4j
10 @Controller
11 // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12 @RequestMapping(value = "/css")
13 public class EightCssController {
```

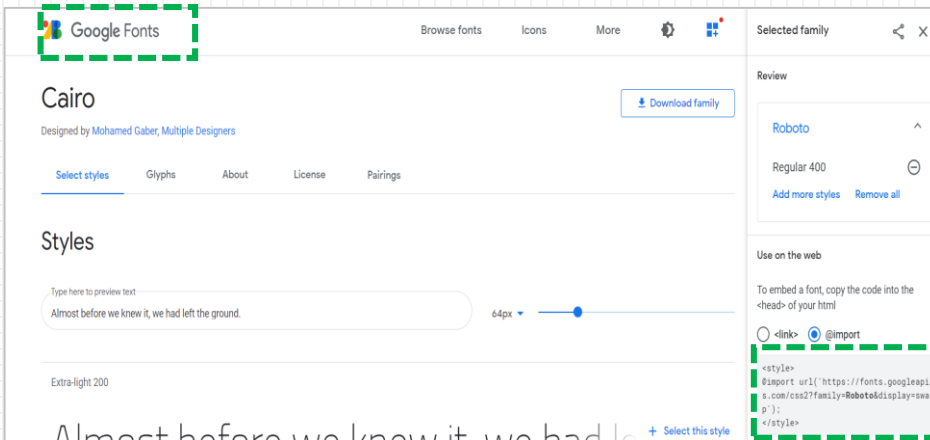
```
79 @GetMapping("/fontTest")
80 public String getFontTest() {
81     log.info("getFontTest()");
82
83     return "csstest/eighth/fontTest";
84 }
```

CSS 위치



CSS

```
1 @import url('https://fonts.googleapis.com/css2?family=Roboto:wght@700&display=swap');
2 .ng-font {
3     font-family: 'Roboto', sans-serif;
4 }
5 p {
6     font-size: 30px;
7 }
```



생성된 html

localhost:7777/css/fontTest

Browser Basic Font, Kaushan Basic Font

Browser Basic Font, Kaushan Basic Font

html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <link rel="stylesheet" href="/css/basic/font.css">
7
8 </head>
9 <body>
10     <p> Browser Basic Font, Kaushan Basic Font </p>
11     <p class="ng-font">Browser Basic Font, Kaushan Basic Font</p>
12 </body>
13 </html>
```

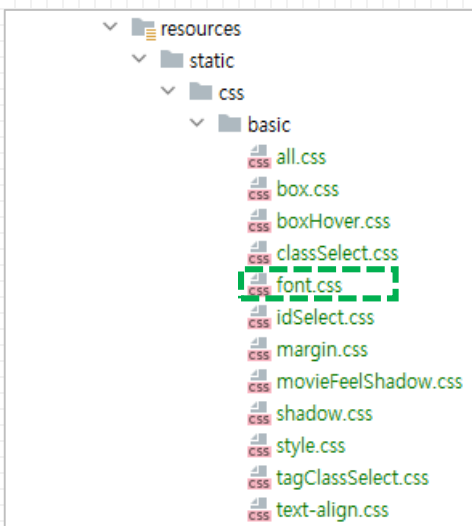
CSS – Shadow

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85  @GetMapping("/shadowTest")
86  public String getShadowTest() {
87      log.info("getShadowTest()");
88
89
90      return "csstest/eighth/shadowTest";
91  }
```

CSS 위치



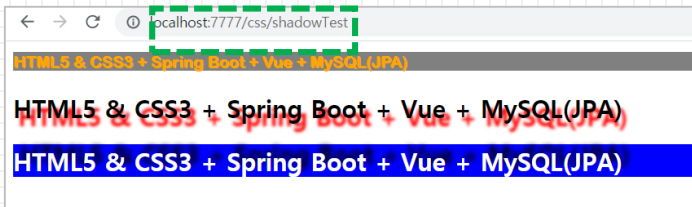
CSS

```
1  h1 {
2      font-size: 16px;
3      font-family: "Arial Rounded MT";
4  }
5
6  .shadow {
7      color: orange;
8      background: gray;
9      text-shadow: 1px 1px;
10 }
11
12 .rb-shadow {
13     text-shadow: 5px 8px 3px #ff0000; /* R G B */
14 }
15
16 .bwb-shadow {
17     background-color: blue;
18     color: #ffffff;
19     text-shadow: 7px -7px 5px #000000;
20 }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/shadow.css">
7  </head>
8  <body>
9      <h1 class="shadow">HTML5 & CSS3 + Spring Boot + Vue + MySQL(JPA)</h1>
10     <h2 class="rb-shadow">HTML5 & CSS3 + Spring Boot + Vue + MySQL(JPA)</h2>
11     <h2 class="bwb-shadow">HTML5 & CSS3 + Spring Boot + Vue + MySQL(JPA)</h2>
12 </body>
13 </html>
```

생성된 html



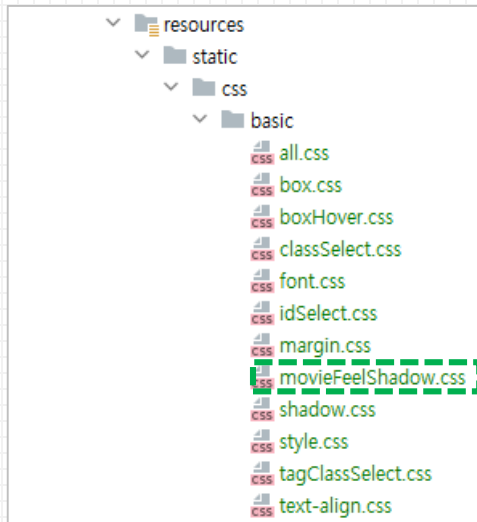
CSS – Shadow2

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
    ...
93  @GetMapping("/movieFeelShadowTest")
94  public String getMovieFeelShadowTest() {
95      log.info("getMovieFeelShadowTest()");
96
97      return "csstest/eighth/movieFeelShadowTest";
98  }
```

CSS 위치



CSS

```
1  body {
2      background: #222222;
3  }
4  h1 {
5      font-size: 80px;
6      font-family: "Arial Rounded MT";
7      letter-spacing: 4px;          /* 자간 */
8  }
9  .movieFeelShadow {
10     color: #000000;
11     text-shadow: 0 0 4px #cccccc, 0 -5px 4px #ffff33, 2px -10px 6px #ffdd33,
12                 -2px -15px 11px #ff8800, 2px -19px 18px #ff2200;
13     /* 여러 색상을 가진 그림자들을 합친다. */
14 }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link rel="stylesheet" href="/css/basic/movieFeelShadow.css">
7  </head>
8  <body>
9      <h1 class="movieFeelShadow">HTML5</h1>
10 </body>
11 </html>
```

생성된 html





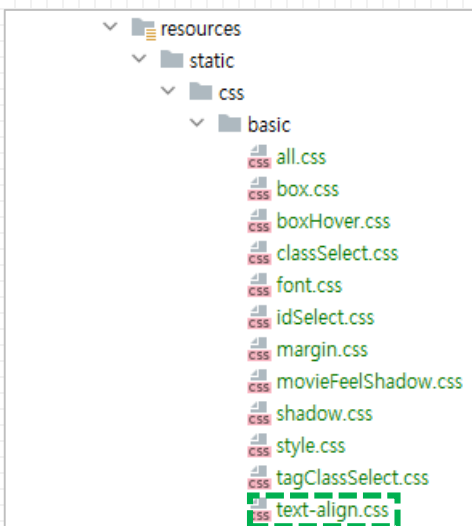
CSS – Text-align

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

```
9  @Slf4j
10  @Controller
11  // 해당 컨트롤러를 사용할 때 무조건 URL 위치가 기본적으로 /css가 들어감
12  @RequestMapping(value = "/css")
13  public class EightCssController {
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100  @GetMapping("/textAlignTest")
101  public String getTextAlignTest() {
102      log.info("getTextAlignTest()");
103
104      return "csstest/eighth/textAlignTest";
105  }
106  }
```

CSS 위치



CSS

```
1  p {
2      border: 1px solid #cccccc;
3      padding: 10px;          /* padding 테두리 */
4      margin: 10px;
5  }
6  .align-left {
7      text-align: left;
8  }
9  .align-right {
10     text-align: right;
11 }
12 .align-center {
13     text-align: center;    /* 가운데 정렬 */
14 }
15 .align-justify {
16     text-align: justify;   /* 양쪽 끝점에서 부터의 가운데, 양쪽 정렬 */
17 }
```

CSS – Text-align

링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6   <link rel="stylesheet" href="/css/basic/text-align.css">
7 </head>
8 <body>
9   <h1>텍스트 정렬 기법</h1>
10  <p class="align-left">
11    오늘도 간다. 내일도 간다. 모레도 간다. 일주일이 간다. 한 달이 간다.
12  </p>
13  <p class="align-right">
14    오늘도 간다. 내일도 간다. 모레도 간다. 일주일이 간다. 한 달이 간다.
15  </p>
16
17  <!-- 내용이 짧을 경우엔 center가 문제가 없음 -->
18  <p class="align-center">
19    오늘도 간다. 내일도 간다. 모레도 간다. 일주일이 간다. 한 달이 간다.
20  </p>
21
22  <!-- 내용이 전체 윤곽을 벗어날 경우 오작동이 발생함 -->
23  <p class="align-center">
24    오늘도 간다. 내일도 간다. 모레도 간다. 일주일이 간다. 한 달이 간다. 일년이 간다. 백년이 간다.
25    천년이 간다. 만년이 간다. 십만년이 간다. 백만년이 간다. 천만년이 간다. 억년이 간다. 십억년이 간다. 백억년이 간다.
26  </p>
27
28  <!-- 내용이 전체 윤곽을 벗어나더라도 양쪽 사이즈에 맞춰 올바른 간격을 맞춤
29    특히 내용이 길어서 윤곽을 벗어나는 상황에서 매우 유용함 -->
30  <p class="align-justify">
31    오늘도 간다. 내일도 간다. 모레도 간다. 일주일이 간다. 한 달이 간다. 일년이 간다. 백년이 간다.
32    천년이 간다. 만년이 간다. 십만년이 간다. 백만년이 간다. 천만년이 간다. 억년이 간다. 십억년이 간다. 백억년이 간다.
33  </p>
34 </body>
35 </html>
```

생성된 html

