

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

JSON

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

컨트롤러 클래스

```
12 @Slf4j
13 // Controller vs RestController의 차이점은 딱 하나다.
14 // Controller는 HTML 을 파싱하는 반면,
15 // RestController는 json을 파싱한다.
16 @RestController
17 public class SixthController {
18
19     @GetMapping("/jsonTest")
20     public String getJsonTest() {
21
22         return "I'm JSON!!!";
23     }
24 }
```

생성된 html

← → ↺ ⓘ localhost:7777/jsonTest

I'm JSON!!!

컨트롤러 클래스

```
25 @GetMapping("/jsonKeyValueTest")
26 public String getJsonKeyValueTest() {
27     // 모든 json 데이터는 아래와 같이 생겼음
28     // 향후 Vue에서 보내는 데이터도 전부 이 json 형태로 날아옴
29     // 우리가 SRT API나 여러가지 Rest API라고 하는 녀석들도
30     // 요청을 하면 데이터 형식이 전부 json 형식으로 날아옴
31
32     String jsonString = "{ \"title\": \"hihi\", \"draft\": false, \"star\": 5 }";
33
34     // JSON 파싱
35     JSONObject jsonObj = new JSONObject(jsonString);
36     String title = jsonObj.getString( key: "title");
37     Boolean draft = jsonObj.getBoolean( key: "draft");
38     Integer star = jsonObj.getInt( key: "star");
39
40     log.info("title: " + title + ", draft: " + draft + ", star: " + star);
41
42     return jsonString;
43 }
```

생성된 html

← → ↺ ⓘ localhost:7777/jsonKeyValueTest

{"title": "hihi", "draft": false, "star": 5}

JSON Viewer Pro

```
{
  "title": "hihi",
  "draft": false,
  "star": 5
}
```

JSON

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

컨트롤러 클래스

```
48 @GetMapping("/jsonMultiObjectTest")
49 public String getJsonMultiObjectTest() {
50     String jsonString = "{" +
51         "\"movie1\": {" +
52             "\"title\": \"hihi\", " +
53             "\"draft\": false, " +
54             "\"star\": 5" +
55         "}, " +
56         "\"movie2\": {" +
57             "\"title\": \"code monkey\", " +
58             "\"draft\": false, " +
59             "\"star\": 5" +
60         "}" +
61     "};";
62
63     // json 내에 Object 형식이 구성된 경우의 파싱
64     JSONObject jsonObj = new JSONObject(jsonString);
65
66     JSONObject movie1obj = jsonObj.getJSONObject("movie1");
67     log.info("movie1: " + movie1obj.toString() +
68         "\ntitle: " + movie1obj.getString(key: "title") +
69         ", draft: " + movie1obj.getBoolean(key: "draft") +
70         ", star: " + movie1obj.getInt(key: "star"));
71
72     JSONObject movie2obj = jsonObj.getJSONObject("movie2");
73     log.info("movie2: " + movie2obj.toString() +
74         "\ntitle: " + movie2obj.getString(key: "title") +
75         ", draft: " + movie2obj.getBoolean(key: "draft") +
76         ", star: " + movie2obj.getInt(key: "star"));
77
78     return jsonString;
79 }
```

생성된 html

← → ↺ ⓘ localhost:7777/jsonMultiObjectTest

```
{"movie1": {"title": "hihi", "draft": false, "star": 5}, "movie2": {"title": "code monkey", "draft": false, "star": 5}}
```

JSON Viewer Pro

← → ↺ JSON Viewer Pro | chrom

```
{
  "movie1": {
    "title": "hihi",
    "draft": false,
    "star": 5
  },
  "movie2": {
    "title": "code monkey",
    "draft": false,
    "star": 5
  }
}
```

JSON

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

컨트롤러 클래스

```
88 @GetMapping("/jsjsonArrayTest")
89 public String getJsonArrayTest() {
90     String jsonString = "{" +
91         "\"movies\": [" +
92         "{" +
93             "\"title\": \"hihi\", " +
94             "\"draft\": false, " +
95             "\"star\": 5" +
96         "}, " +
97         "{" +
98             "\"title\": \"code monkey\", " +
99             "\"draft\": false, " +
100             "\"star\": 5" +
101         "}, " +
102         "{" +
103             "\"title\": \"monkey magic\", " +
104             "\"draft\": false, " +
105             "\"star\": 4.7" +
106         "}" +
107         "]" +
108         "}";
109
110     // JSON 배열 파싱
111     JSONObject jsonObj = new JSONObject(jsonString);
112     JSONArray jArr = jsonObj.getJSONArray( key: "movies");
113
114     // 루프를 돌며 JSON 배열의 모든 정보를 출력함
115     for (int i = 0; i < jArr.length(); i++) {
116         JSONObject obj = jsonObj.getJSONObject(i);
117
118         String title = obj.getString( key: "title");
119         Boolean draft = obj.getBoolean( key: "draft");
120         Float star = obj.getFloat( key: "star");
121
122         log.info("title: " + title + ", draft: " + draft + ", star: " + star);
123     }
124     return jsonString;
125 }
126 }
```

생성된 html

```
< > ↺ localhost:7777/jsjsonArrayTest

{"movies": [{"title": "hihi", "draft": false, "star": 5}, {"title": "code monkey", "draft": false, "star": 5}, {"title": "monkey magic", "draft": false, "star": 4.7}]}
```

JSON Viewer Pro

```
{
  "movies": [
    {
      "title": "hihi",
      "draft": false,
      "star": 5
    },
    {
      "title": "code monkey",
      "draft": false,
      "star": 5
    },
    {
      "title": "monkey magic",
      "draft": false,
      "star": 4.7
    }
  ]
}
```

```
// 현재 시점까지의 json 처리는
// 향후 Vue에서 AXIOS라는 것을 사용하며 모두 처리가 될 것임
// 한 가지 차이가 있다면 나중에 python과 연동할 때
// Spring의 Requester를 통해서
// Spring 자체가 Client가 되어 Python Server에 요청을 보내야함
// 그 시점에서는 Spring이 직접 JSON을 처리할 필요가 있음
```

JavaScript

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

컨트롤러 클래스

```
7 @Slf4j
8 @Controller
9 public class EleventhJavaScriptTestController {
10
11     @GetMapping("/localeStringTest")
12     public String getLocaleStringTest() {
13         log.info("getLocaleStringTest()");
14
15         return "/javascript/eleventh/localeStringTest";
16     }
17 }
```

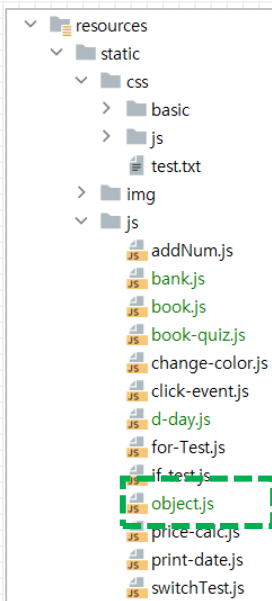
html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <link rel="stylesheet" href="/css/js/obj.css">
7 </head>
8 <body>
9     <div id="display"></div>
10    <script src="/js/object.js"></script>
11 </body>
12 </html>
```

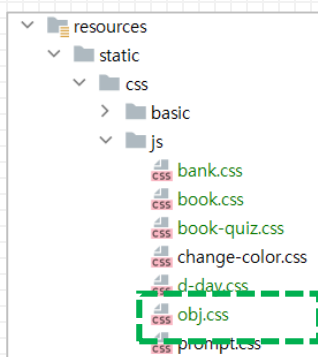
css

```
1 #display {
2     width: 500px;
3     margin: 20px auto;
4     padding: 10px;
5     border: 1px solid #ccc;
6     text-align: center;
7     font-size: 20px;
8     line-height: 30px;
9 }
```

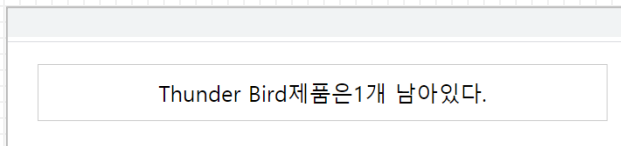
.js 위치



.css 위치



생성된 html



js

```
1 // object = ? 객체
2 // 객체는 ? 데이터가 메모리에 올라간 상태
3 // 자바 스크립트에서는 별도로 클래스나 특정 키워드 없이도
4 // 아래와 같이 중괄호로 묶어버리면 클래스 형태의 객체를 만들 수 있다.
5 var toyRobot = {
6     // 형태가 json 형태를 따르고 있다.
7     // 자바스크립트 내부에서 값을 배치하는 작업은 JSON 형식을 따른다.
8     productId: "123-34",
9     name: "Thunder Bird",
10    price: "25,000,000",
11    madeIn: "Korea",
12    quantity: 1,
13    showStock: function () {
14        document.querySelector('#display').innerHTML =
15            this.name + "제품은" + this.quantity + "개 남아있다.";
16    }
17 };
18
19 toyRobot.showStock();
```



JavaScript - function

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

컨트롤러 클래스

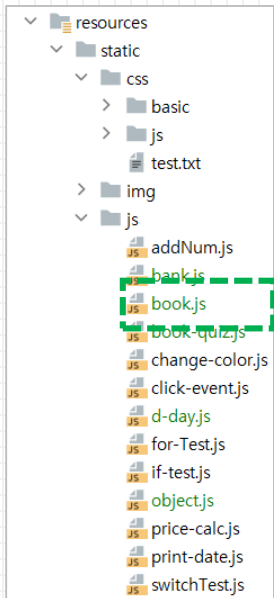
```
7  @Slf4j
8  @Controller
9  public class EleventhJavaScriptTestController {

18     @GetMapping("/customObjectTest")
19     public String getCustomObjectTest() {
20         log.info("getCustomObjectTest()");
21
22         return "/javascript/eleventh/customObjectTest";
23     }
}
```

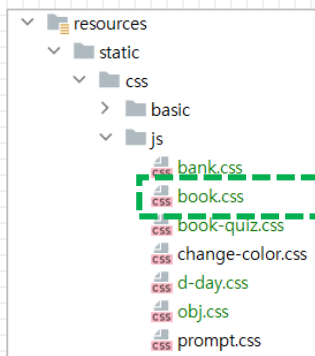
html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <link rel="stylesheet" href="/css/js/book.css">
6  </head>
7  <body>
8      <div id="display"></div>
9      <script src="/js/book.js"></script>
10 </body>
11 </html>
```

.js 위치



.css 위치



CSS

```
1  body {
2      /* background:url('/img/sale.png') repeat-x top left; */
3      background:url('/img/sale.png') repeat-x top;
4  }
5  h1 {
6      text-align: center;
7      margin-top: 400px;
8  }
9  p {
10     text-align: center;
11     width: 98%;
12     margin-bottom: 10px;
13     padding: 10px;
14     border: 1px solid #cccccc;
15 }
```

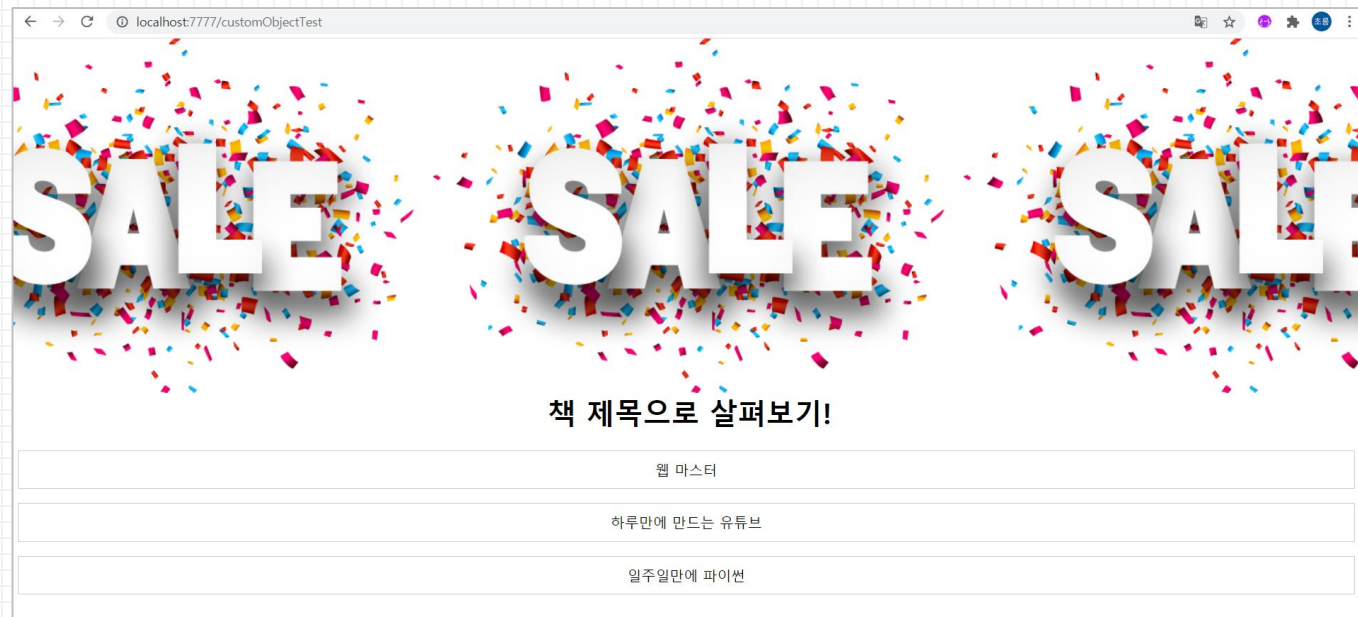
JavaScript - function

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

js

```
1 // function : 일종의 생성자 역할을 해줄 수 있는 함수
2 function Book (title, author, volume, price) {
3     this.title = title;
4     this.author = author;
5     this.volume = volume;
6     this.price = price;
7 }
8
9 // 작성한 함수를 사용해서 html, youtube, python 객체를 만든다.
10 var html = new Book ('웹 마스터', 'Ko', '608', '38,000');
11 var youtube = new Book ('하루만에 만드는 유튜브', 'Kim', '1068', '40000');
12 var python = new Book ('일주일만에 파이썬', 'Park', '352', '18000');
13 // ex.python 객체를 열어서 까보면 JSON과 같이 배치되어 있다.
14
15 // html, youtube, python 객체를 배열에 저장
16 var bookList = [html, youtube, python];
17
18 // bookList: [ {}, {}, {} ]
19 // html : { title: ~~~, author: ~~~, volume: ~~~, price: ~~~ }
20
21 document.write ("<h1>책 제목으로 살펴보기!</h1>");
22
23 for ( var i =0; i < bookList.length; i++) {
24     document.write("<p>"+ bookList[i].title +"</p>");
25 }
```

생성된 html





JavaScript - calcDate

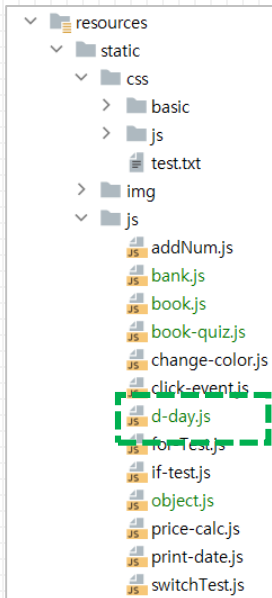
링크 <https://github.com/limcholong/LectureContents/tree/main/javascript/CholongLim/cholongtest/src/main>

컨트롤러 클래스

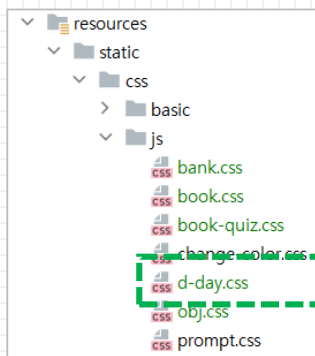
```
7  @Slf4j
8  @Controller
9  public class EleventhJavaScriptTestController {

33     @GetMapping("/destDayTest")
34     public String getDestDayTest() {
35         log.info("getDestDayTest()");
36
37         return "/javascript/eleventh/destDayTest";
38     }
}
```

.js 위치



.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/js/d-day.css">
7  </head>
8  <body>
9      <div class="container">
10         <div class="day1">
11             <h3>만난지</h3>
12             <p id="accent" class="accent"><span style="font-style: italic">몇 일 ?</span> </p>
13         </div>
14         <div class="bar">기념일 계산</div>
15         <div class="day2">
16             <ul>
17                 <li class="item-title">100일</li>
18                 <li class="item-date" id="date100"></li>
19             </ul>
20             <ul>
21                 <li class="item-title">200일</li>
22                 <li class="item-date" id="date200"></li>
23             </ul>
24             <ul>
25                 <li class="item-title">1년</li>
26                 <li class="item-date" id="date365"></li>
27             </ul>
28             <ul>
29                 <li class="item-title">500일</li>
30                 <li class="item-date" id="date500"></li>
31             </ul>
32         </div>
33     </div>
34     <script src="/js/d-day.js"></script>
35 </body>
36 </html>
```


JavaScript - calcDate

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

CSS

```
1 @import url('https://fonts.googleapis.com/css?family=Black+Han+Sans|Jua');
2 * {
3   box-sizing: border-box;
4 }
```

```
5 .container {
6   width: 450px;
7   margin: 0 auto;
8   border: 1px solid #cccccc;
9   border-radius: 2%;
10  box-shadow: 2px 2px 5px #333333;
11 }
12
13 .day1 {
14   padding-top: 20px;
15   text-align: center;
16 }
17
18 .day1 h3 {
19   font-size: 1.2em;
20   color: #666666;
21 }
22
23 .accent {
24   margin-left: 10px;
25   margin-right: 10px;
26   margin-top: 10px;
27   font-family: 'Jua', sans-serif;
28   font-weight: bold;
29   font-size: 3.5em;
30   color: #222222;
31 }
```

```
33 .bar {
34   width: 100%;
35   margin: 60px auto 0 auto;
36   padding-left: 15px;
37   height: 40px;
38   background: #747474;
39   color: #ffffff;
40   font-size: 1.2em;
41   line-height: 40px;
42 }
43
44 .day2 {
45   width: 420px;
46   margin: 20px auto 20px auto;
47 }
48
49 .day2 ul {
50   list-style: none;
51   border-bottom: 1px dashed #cccccc;
52   height: 60px;
53 }
54
55 .day2 ul:last-child {
56   border-bottom: none;
57 }
```

```
59 .item-title {
60   float: left;
61   width: 160px;
62   font-weight: bold;
63   font-size: 1.5em;
64   line-height: 60px;
65 }
66
67 .item-date {
68   float: left;
69   margin-left: 60px;
70   font-size: 1.2em;
71   color: #222222;
72   text-align: right;
73   line-height: 60px;
74 }
```

JavaScript - calcDate

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

js

```
1 // 오늘 날짜 정보
2 var now = new Date();
3 // 처음 만난 날짜 정보
4 var firstDay = new Date("2008-10-23");
5 // 오늘 날짜 정보를, 컴퓨터가 계산하는 숫자 정보로 변환
6 // 정확하게는 ms 단위로 바꾼다.
7 var toNow = now.getTime();
8 // 처음 만난 날짜 정보를, 컴퓨터가 계산하는 숫자 정보로 변환
9 var toFirst = firstDay.getTime();
10 // 오늘 날짜 - 첫 만남
11 var passedTime = toNow - toFirst;
12 // 다시 날짜로 변환
13 // Math.round(반올림)
14 var passedDay = Math.round(passedTime / (24 * 60 * 60 * 1000));
15
16 document.querySelector('#accent').innerText = passedDay + "일";
17
18 calcDate(100);
19 calcDate(200);
20 calcDate(365);
21 calcDate(500);
22
23 function calcDate (days) {
24     var future = toFirst + days * (1000 * 60 * 60 * 24 );
25     var someday = new Date(future);
26
27     var year = someday.getFullYear();
28     var month = someday.getMonth() + 1;
29     var date = someday.getDate();
30
31     document.querySelector('#date' + days).innerText =
32         year + "년" + month + "월" + date + "일";
33 }
```

생성된 html

만난지	
4631일	
기념일 계산	
100일	2009년1월31일
200일	2009년5월11일
1년	2009년10월23일
500일	2010년3월7일