

[디지털 컨버전스] 스마트 콘텐츠와 웹 융합 응용SW 개발자 양성과정

강사 : 이상훈

학생 : 임초롱

HTML

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

```
10 @Slf4j
11 @Controller
12 public class HtmlTestController {
13     // HtmlTestController가 제어하는 요소를 디버깅하도록 서포트하는 객체
14     // static final Logger log = LoggerFactory.getLogger(HtmlTestController.class)
15
16     // 실제 웹상에서 URL 요청할 때는 자주 사용하는 두 가지 방식이 있다.
17     // Get, Post 방식인데 우선 일반적인 URL 입력은 Get으로 파악하면 되겠다.
18     // 현재 우리의 ip:port 주소의 Home을 의미한다.
19     // ("/")메인페이지 홈,
20
21     // Controller가 사용자의 'ip:port/' 형식의 요청을 하였을때 동작할 내용 정의
22     @GetMapping("/")
23     public String firstIndex() {
24         // 위에 Logger 사용하는 방식은 Lombok이 없을 경우 사용함
25         // 그래서 이클립스에서는 상당히 불편하게 구성해야하는데 비해
26         // 여기서는 맨 위쪽의 @Slf4j 를 추가함으로써
27         // 위의 Logger 파트를 적어줄 필요가 없어짐
28
29         log.info("firstIndex() 실행");
30
31         // 여기서 리턴되는 String 값은 resources / templates / htmlHi.html을 의미함
32         // 즉 이 부분에 무엇을 배치하느냐에 따라 출력되는 내용이 제각각 달라짐을 의미 !
33         return "htmlHi";
34     }
35 }
```

@Slf4j

static final Logger log =
 LoggerFactory.getLogger(HtmlTestController.class);

위와 같이 적을 필요 없이, @Slf4j 를 추가하여 사용 가능하다.

@GetMapping("")

GetMapping ("") 에서 ""안에 들어가는 값이
html 주소가 된다.

HTML

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

컨트롤러 클래스

```
7  @Slf4j
8  @Controller
9  public class FirstController {
10     @GetMapping("/htmlList")
11     public String doHtmlList() {
12         log.info("doHtmlList()");
13
14         return "first/list";
15     }
16 }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      <!-- h1 ~ h6는 글자의 크기 속성을 지정할 수 있음. -->
9      <h1>Frontend/Backend</h1>
10     <p>웹 개발에는 Frontend/Backend 개발이 있고 </p>
11     <p>두 개를 모두 하는 경우 Full Stack 개발자라고 부릅니다.</p>
12     <p>Full Stack 개발자의 필수 덕목은 Spring Boot, JPA + Alpha(Kafka, ets ...)</p>
13     <p>그리고 HTML5/CSS3, JavaScript, Vue/React/Svelte, Typescript 등입니다.</p>
14
15     <!-- unordered list : 순서가 없다. -->
16     <ul>
17         <li>HTML5/CSS3</li>
18         <li>JavaScript</li>
19         <li>Vue</li>
20         <li>Svelte</li>
21         test<br>
22     </ul>
23
24     <!-- ordered list : 순서가 있다. -->
25     <ol>
26         <li>Golang</li>
27         <li>c#</li>
28     </ol>
29 </body>
30 </html>
```

생성된 html

localhost:7777/htmlList

Frontend/Backend h1

웹 개발에는 Frontend/Backend 개발이 있고
두 개를 모두 하는 경우 Full Stack 개발자라고 부릅니다.
Full Stack 개발자의 필수 덕목은 Spring Boot, JPA + Alpha(Kafka, ets ...)
그리고 HTML5/CSS3, JavaScript, Vue/React/Svelte, Typescript 등입니다.

- HTML5/CSS3
 - JavaScript
 - Vue
 - Svelte
 - test
- Golang
 - c#

h1 ~ h6 :

글자 크기 속성 지정한다.

ul :

Unordered list, 순서가 없다.

ol :

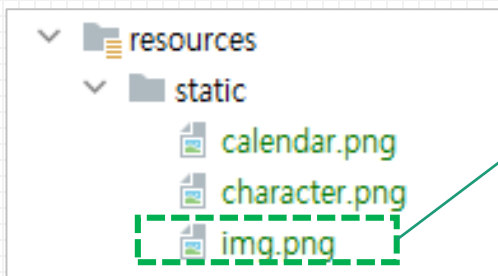
ordered list, 순서가 있다.

HTML – image

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

컨트롤러 클래스

```
7  @Slf4j
8  @Controller
9  public class FirstController {
17     @GetMapping("/imageTest")
18     public String doimageLTest() {
19         log.info("doimageTest()");
20
21         return "first/image";
22     }
}
```



html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      <h1>고양미란...</h1>
9      
10     <p>고양미 ~ </p>
11
12 </body>
13 </html>
```

** :**

이미지 지정 시 <body>내부에 위와 같이 작성한다.

이 경우 사용되는 사진의 파일명이 img.png 였기때문에 이다.



HTML – font

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

컨트롤러 클래스

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

24  @GetMapping("/fontTest")
25  public String dofontTest() {
26      log.info("dofontTest()");
27
28      return "first/font";
29  }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <!-- style 에 적용하면 전체에 적용됨 -->
7      <style>
8          p {
9              font-size: 32px; /*글자 크기*/
10             line-height: 45px /*줄 간격*/
11         }
12     </style>
13 </head>
14 <body>
15     <h2>Go Language</h2>
16     <!--Strong: 강조, b: 또한 강조 -->
17     <p><strong>Rust</strong>, c, <b>Java</b>, Python</p>
18     <!--em: 이탤릭체 강조, i: 이탤릭체 -->
19     <p><em>Lisp</em>, Fortran, <i>JavaScript</i>, asm<br>
20         FPGA Verilog</p>
21 </body>
22 </html>
```

생성된 html

localhost:7777/fontTest

Go Language

Rust, c, Java, Python

Lisp, Fortran, JavaScript, asm

FPGA Verilog

** :**
강조 표시 (글씨체가 두꺼워진다.)

** :**
강조 표시 (글씨체가 두꺼워진다.)

** :**
이탤릭체 강조 표시

<i></i> :
이탤릭체

<style></style> :
style 내부에 적용하면 전체에 적용된다.

```
<style>
  p {
    font-size: 32px; /*글자 크기*/
    line-height: 45px /*줄 간격*/
  }
</style>
```

P들의 글자 크기, 줄 간격을 설정한다.

HTML - mark

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

컨트롤러 클래스

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

31  @GetMapping("/markTest")
32  public String domarkTest() {
33      log.info("domarkTest()");
34
35      return "first/mark";
36  }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      <!-- mark : 형광펜 기능 = 디폴트 옐로우 -->
9      <h2>Python Language: <mark style="color: red;">TensorFlow" ==> "Keras"</mark></h2>
10     <!-- span: 줄바꿈 없이 영역을 묶음, style="color: x" 특정 색상으로 만들 -->
11     <p>Machine Learning, <span style="color: blue;">Deep Learning,
12     SLAM, Signal Processing, </span>Statistics </p>
13     <!-- background-color: x : 배경색을 지정하고 싶다면 -->
14     <h3>Test <mark style="background-color: green;">Driven Development</mark> </h3>
15 </body>
16 </html>
```

<mark></mark> :

형광펜 기능을 하며 기본 색상은 옐로우 이다.

** :**

span은 줄바꿈 없이 영역을 묶는다.
style = "color : x;" 글자색을 특정 색상으로 만든다.

<mark>" background-color: x;" </mark> :

배경색을 지정할 수 있다.
style = "color : x" 특정 색상으로 만든다.

생성된 html

← → ↻ ⓘ localhost:7777/markTest

mark style = "color: red"

Python Language: "TensorFlow" ==> "Keras"

span style = "color: blue"

Machine Learning, Deep Learning, SLAM, Signal Processing, Statistics

Test Driven Development

mark style = "background color: green;"

HTML – table1

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

컨트롤러 클래스

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

38  @GetMapping("/firsttableTest")
39  public String firsttableTest() {
40      log.info("firsttableTest()");
41
42      return "first/tableTest1";
43  }
```

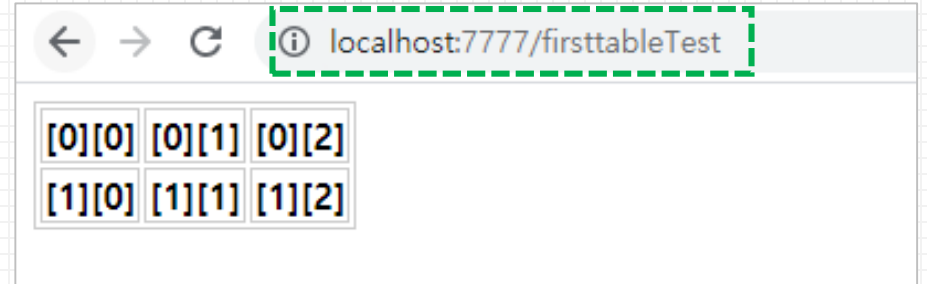
Padding :

셀 테두리와 내용 사이의 간격.

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  <style>
7      table, th, td{
8          border: 1px solid #cccccc;
9      }
10     td{
11         padding: 5px;
12     }
13 </style>
14 </head>
15 <body>
16 <table>
17     <!-- tr : 행을 뜻함)-->
18     <tr>
19         <!-- th : 열을 뜻함 (현재 3개)-->
20         <th>[0][0]</th>
21         <th>[0][1]</th>
22         <th>[0][2]</th>
23     </tr>
24     <tr>
25         <th>[1][0]</th>
26         <th>[1][1]</th>
27         <th>[1][2]</th>
28     </tr>
29 </table>
30 <!-- br : enter (한 줄 띄움)-->
31 </body>
32 </html>
```

생성된 html



<th></th> :

table head의 약자로 표의 제목을 쓰는 역할을 한다.
(기본 값은 굵은 글씨체, 중앙 정렬)

<tr></tr> :

table row의 약자로 가로줄을 만드는 역할을 한다.
(기본 값은 보통 글씨체, 왼쪽 정렬)

<td></td> :

table data의 약자로 셀을 만드는 역할을 한다.
(기본 값은 보통 글씨체, 왼쪽 정렬)

HTML – table2

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

컨트롤러 클래스

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

45  @GetMapping("/secondtableTest")
46  public String secondtableTest() {
47      log.info("secondtableTest()");
48
49      return "first/tableTest2";
50  }
```

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  <style>
7      table, th, td {
8          border: 1px solid #000000;
9      }
10     th{
11         padding: 15px; /*셀 테두리와 내용 사이의 간격(패딩)*/
12     }
13     tr > td:nth-child(odd){
14         width: 120px; /* 홀수번째 열의 너비 지정 */
15     }
16     tr > td:nth-child(even) {
17         width: 300px; /* 짝수번째 열의 너비 지정 */
18     }
19 </style>
20 </head>
```

```
21 <body>
22 <table>
23     <tr>
24         <!-- th는 td의 강조형임-->
25         <th>Name</th>
26         <td></td>
27         <th>Contact</th>
28         <td></td>
29     </tr>
30     <tr>
31         <th>Address</th>
32         <!-- colspan="3"은 열 3개를 하나로 묶음-->
33         <td colspan="3"></td>
34     </tr>
35     <tr>
36         <th>Self Introduction</th>
37         <td colspan="3"></td>
38     </tr>
39 </table>
40 </body>
41 </html>
```

생성된 html

← → ↻ ⓘ localhost:7777/secondtableTest

Name		Contact	
Address			
Self Introduction			

<colspan = "3" > :
열 3개를 하나로 묶는다.

HTML – table3

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

컨트롤러 클래스

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

52  @GetMapping("/thirdtableTest")
53  public String thirdtableTest() {
54      log.info("thirdtableTest()");
55
56      return "first/tableTest3";
57  }
```

생성된 html

← → ↻ ⓘ localhost:7777/thirdtableTest

이력서(Resume)

누구도 나를 멈출 수 없다. 다 비켜라 내 안의 흑염룡 ?!

Browser	Vendor	Downloads
Chrome	Google	https://www.google.com/chrome/
Firefox	Mozilla	https://www.mozilla.org/ko/firefox

html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  <style>
7      table, th, td {
8          border: 1px solid #000000;
9      }
10     td, th{
11         padding: 10px;
12     }
13 </style>
14 </head>
15 <body>
16 <table>
17     <!--caption, figcaption을 통해 표에 제목을 붙일 수 있음 -->
18     <caption>
19         <strong>이력서(Resume)</strong>
20         <p>누구도 나를 멈출 수 없다. 다 비켜라 내 안의 흑염룡 ?!</p>
21     </caption>
22 <tr>
23     <th>Browser</th>
24     <th>Vendor</th>
25     <th>Downloads</th>
26 </tr>
27 <tr>
28     <th>Chrome</th>
29     <td>Google</td>
30     <td>https://www.google.com/chrome/</td>
31 </tr>
32 <tr>
33     <th>Firefox</th>
34     <td>Mozilla</td>
35     <td>https://www.mozilla.org/ko/firefox</td>
36 </tr>
37 </table>
38 </body>
39 </html>
```

<caption></caption> :

표에 제목을 붙일 수 있다.

<figcaption></figcaption> :

표에 제목을 붙일 수 있다.