
Chapter 08

유효성 검사: 도서 등록 데이터의 유효성 검사하기

Q & A

- 언제라도 질문하세요
모르면 외우면 되지만 코딩 안되면 꼭 질문

1. 대면수업
2. 한성e-class 질의응답게시판

- 강사의 1번 선생님은 여러분의 질문



목차

1. 유효성 검사의 개요
2. 기본 유효성 검사
3. 데이터 형식 유효성 검사
4. [북마켓] 도서 등록 데이터의 유효성 검사하기

학습목표

1. 나는 유효성 검사의 개념을 이해한다.
2. 나는 유효성 검사를 위한 핸들러 함수의 작성 방법을 익힌다.
3. 나는 유효성 검사를 처리하는 방법을 익힌다.
4. 나는 유효성 검사를 이용하여 도서 상품 등록 데이터를 검증한다.

1층(1장): 쇼핑몰 프로젝트 생성하기

7층(7장): 도서이미지 등록하기

8층(8장): 도서등록 데이터의 유효성 검사하기

Section 01

유효성 검사의 개요

1. 유효성 검사를 위한 핸들러 함수

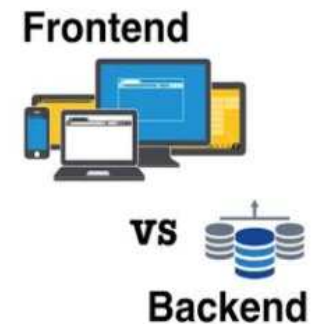
■ 유효성 검사(validation)

- 사용자가 폼 페이지에서 입력한 데이터 값이 서버로 전송되기 전에 특정 규칙에 맞게 입력되었는지 검증하는 것
- 사용자가 실수로 유효하지 않은 데이터 값을 입력하면 부적합하다고 판단하여 다시 폼 페이지로 되돌려 사용자에게 오류가 있음을 알려줌
- 유효성 검사의 예
 - ✓ 입력 데이터가 null인지 확인하는 검사
 - ✓ 폼 페이지에서 나이를 입력할 때 숫자를 인식하는 검사

1. 유효성 검사를 위한 핸들러 함수

■ 유효성 검사를 위한 핸들러 함수

- 핸들러 함수
 - ✓ 폼 페이지에서 이벤트가 발생했을 때의 유효성 검사를 위해 매핑하는 메소드
- 자바스크립트를 이용하여 유효성 검사를 위한 코드 작성
 - ✓ 자바스크립트는 웹 브라우저에서 유효성 검사를 처리함
 - ✓ 따라서 서버에서 처리하는 것보다 속도가 빠르고 서버에 과부하를 주지 않음
- 사용자가 폼 페이지에 입력한 데이터 값이 서버로 전송되기 전에 특정 규칙에 맞게 입력되었는지 검사함
- 입력된 데이터가 유효성 검사를 통과하면 서버로 전송하고, 그렇지 않으면 서버 전송을 취소하고 사용자에게 오류 메시지를 보여주는 역할을 함



1. 유효성 검사를 위한 핸들러 함수

■ 유효성 검사를 위한 핸들러 함수

```
<script type="text/javascript">
    function 핸들러 함수() {
        var str=document.폼이름.입력항목이름.value;
    }
</script>
<form name="폼이름">
    ...생략...
    <input type="submit" onclick="핸들러 함수()">
</form>
```


1. 유효성 검사의 개요

■ 유효성 검사를 위해 핸들러 함수를 만드는 과정

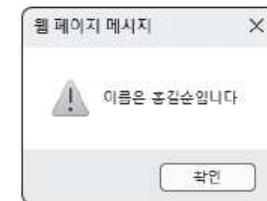
- ❶ input 태그의 type 속성 값이 submit인 경우 onclick 속성을 이용하여 핸들러 함수를 설정 또는 form 태그의 onsubmit 속성 값에 설정
- ❷ 자바스크립트를 이용하여 </script>...<script> 내에 핸들러 함수를 작성
<script>...</script> 구문은 JSP 페이지의 어디에 위치해도 상관 없음
- ❸ 폼 페이지에서 입력된 데이터 값을 핸들러 함수로 가져오기 위해 form 태그의 name 속성 또는 forms 객체 이용
 - ✓ forms 객체를 이용하는 경우 : forms 객체는 배열의 형태이기 때문에 length 속성으로 크기를 알 수 있고 배열 값인 index는 form 태그가 나타나는 순서로 0부터 시작

1. 유효성 검사의 개요

■ 핸들러 함수 사용 예

```
<%@ page contentType="text/html; charset=utf-8" %>
<html>
<head>
<title>Validation</title>
</head>
<script type="text/javascript">
    function checkForm() {
        alert("이름은 " + document.frm.name.value + "입니다");
    }
</script>
<body>
    <form name="frm">
        <p> 이름 : <input type="text" name="name">
            <input type="submit" value="전송" onclick="checkForm()">
        </form>
</body>
</html>
```

이름 : 홍길순



1. 유효성 검사의 개요

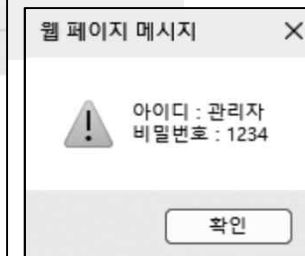
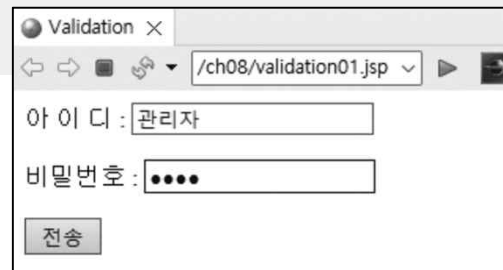
■ [예제 8-1] 폼 페이지에 입력한 아이디와 비밀번호 출력하기

학생> <script type="text/javascript">에서 type이 필요한가요?
웹기초에서는 type 사용하지 않았습니다

교수> 필요없습니다

JSPBook/src/main/webapp/ch08/validation01.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <title>Validation</title>
05 </head>
06 <script type="text/javascript">
07     function checkForm() {
08         alert("아이디 : " + document.loginForm.id.value + "\n" +
09             "비밀번호 : " + document.loginForm.passwd.value);
10     }
11 </script>
12 <body>
13     <form name="loginForm">
14         <p>아 이 디 : <input type="text" name="id">
15         <p>비밀번호 : <input type="password" name="passwd">
16         <p><input type="submit" value="전송" onclick="checkForm()">
17     </form>
18 </body>
19 </html>
```



2. 유효성 검사 처리 방법

■ 유효성 검사 처리 방법

표 8-1 유효성 검사의 종류

유효성 검사	설명
기본 유효성 검사	폼 페이지에 입력된 데이터 값의 존재 유무를 검사한다.
데이터 형식 유효성 검사	폼 페이지에 입력된 데이터 값이 특정 패턴에 적합한지 여부를 검사하며 정규 표현식을 사용한다.

Section 02

기본 유효성 검사

1. 데이터 유무 확인하기

■ 기본 유효성 검사

- 사용자가 폼 페이지의 입력 항목에 입력한 데이터 값이 있는지 없는지 확인하고 데이터 길이, 숫자 등 기본적인 것이 맞는지 검사
- 폼 페이지의 입력 데이터 길이를 확인하여 데이터의 유무를 검증하는 것은 기본 유효성 검사에 해당

■ 데이터 유무 확인하기

- 데이터 값의 유무에 대한 검사
 - ✓ [예] 회원 가입 페이지에서 사용자가 아이디와 비밀번호 등의 필수 입력 항목을 입력하지 않고 <전송>을 클릭하면 입력하지 않았다는 오류 메시지가 나타나는 것
- 입력 데이터의 유무를 검사하는 형식

```
document.폼이름.입력양식이름.value == ""
```

1. 데이터 유무 확인하기

■ 입력 데이터의 유무 검사 예

```
<%@ page contentType="text/html; charset=utf-8" %>
<html>
<head>
<title>Validation</title>
</head>
<script type="text/javascript">
    function checkForm() {
        if (document.frm.name.value=="") {
            alert("이름을 입력해주세요.");
            document.frm.name.select();
        }
    }
</script>
<body>
    <form name="frm">
        <p>이름 : <input type="text" name="name">
            <input type="submit" value="전송" onclick="checkForm()">
        </form>
</body>
</html>
```

이름 :

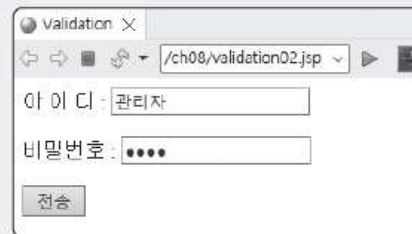


1. 데이터 유무 확인하기

■ [예제 8-2] 폼 페이지에 입력한 데이터(아이디와 비밀번호) 값의 유무 검사하기

JSPBook/src/main/webapp/ch08/validation02.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <title>Validation</title>
05 </head>
06 <script type="text/javascript">
07     function checkLogin() {
08         var form=document.loginForm;
09         if (form.id.value=="") {
10             alert("아이디를 입력해주세요.");
11             form.id.focus();
12             return false;
13         } else if (form.passwd.value=="") {
14             alert("비밀번호를 입력해주세요.");
15             form.passwd.focus();
16             return false;
17         }
18         form.submit();
19     }
20 </script>
21 <body>
```



```
22 <form name="loginForm" action="validation02_process.jsp" method=
    "post">
23     <p> 아 이 디 : <input type="text" name="id">
24     <p> 비밀번호 : <input type="password" name="passwd">
25     <p> <input type="button" value="전송" onclick="checkLogin()">
26 </form>
27 </body>
28 </html>
```


1. 데이터 유무 확인하기

교수> 6장 form04.jsp와 8장 validation02.jsp를 비교합니다.

학생> 왜요?

교수> 변경사항을 보고자 합니다.

Form04.jsp 26줄에서 <input type="submit" value="가입하기"> submit버튼 클릭하면 form04_process.jsp로 form값이 전달됩니다

Validation02.jsp 25번줄에서 <input type="button" value="전송" onclick="checkLogin()" > 버튼 클릭하면 checkLogin() 이벤트 리스너 실행

학생> validation02.jsp에서 form값 전달은 어디에서 되나요?

교수> 18번줄 form.submit()에서 validation02_process.jsp로 form값이 전달됩니다.

Submit() 메소드에 대한 설명은 어디에 있나요? 웹기초에서 배우지 않았습니까.

W3schools.com에서 submit 검색

https://www.w3schools.com/jsref/met_form_submit.asp

학생> 8번줄, 9번줄에서 HTML 태그의 속성값을 가져오네요.

11번줄 focus() 메소드는

W3schools.com에서 focus 검색

https://www.w3schools.com/jsref/met_html_focus.asp

학생> 12번줄에서 return false; 하면 25번줄에서 onclick 대기하겠네요

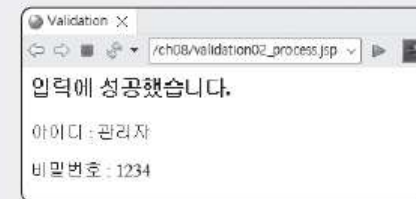
정상적으로 입력하고 onclick하면 18번줄에서 submit()메소드로 validation02_process.jsp로 form값이 전달되겠네요.

1. 데이터 유무 확인하기

■ [예제 8-2] 폼 페이지에 입력한 데이터(아이디와 비밀번호) 값의 유무 검사하기

JSPBook/src/main/webapp/ch08/validation02_process.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <title>Validation</title>
05 </head>
06 <body>
07     <h3>입력에 성공했습니다.</h3>
08     <%
09         request.setCharacterEncoding("utf-8");
10         String id=request.getParameter("id");
11         String passwd=request.getParameter("passwd");
12     %>
13     <p> 아이디 : <%= id %>
14     <p> 비밀번호 : <%= passwd %>
15 </body>
16 </html>
```



1. 유효성 검사의 개요

■ [예제 8-2] JS 디버깅

교수> 웹기초 6장 28쪽과 같이 JS 디버깅을 합니다.

학생> 왜요?

교수> JS는 크롬에서 실행되므로 디버깅은 eclipse가 아닌 F12로 디버깅합니다.

- 1) 8장 validation02.jsp 실행
- 2) 크롬에서 8번줄에 중단점설정
- 3) 아이디와 비밀번호입력
- 4) 중단점도착하면 Local창에서 form변수가 undefined 확인
- 5) Step over 버튼한후 Local창에서 form.id.value 값 확인
- 6) 중단점해제
- 7) Resume 버튼

학생> Local창 form.id에서 value가 안보입니다.

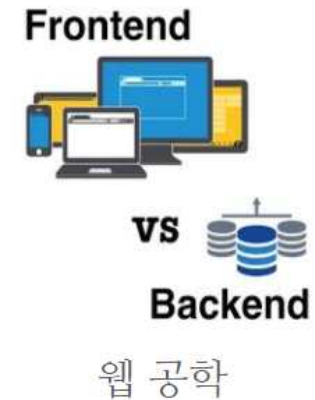
교수> (...)을 클릭합니다

```
onsecuritypolicy
onseeked: null
(...)
inert: false
innerHTML: "<div></div>"
innerText: "ID ID"
```

학생> 과제8에서 JSP디버거

과제11에서 Java파일 디버깅

과제15에서 JS코드 디버깅해 보니 각 챕터별 연습문제와 시험문제에서 많이 활용하겠습니다



2. 데이터 길이 확인하기

■ 데이터 길이 확인하기

- 회원 가입 페이지에서 아이디, 비밀번호 등과 같은 입력 데이터의 제한 길이를 검사하는 것
 - ✓ [예] 입력 데이터의 조건으로 아이디와 비밀번호는 4~12자 이내로 영어와 숫자를 혼합해서 입력할 것, 첫 문자는 숫자로 시작할 수 없음 등을 검사

```
document.폼이름.입력양식이름.value.length
```

2. 데이터 길이 확인하기

■ 입력 데이터 길이 검사 예

```
<%@ page contentType="text/html; charset=utf-8" %>
<html>
<head>
<title>Validation</title>
</head>
<script type="text/javascript">
    function checkForm() {
        if (document.frm.name.value.length < 6 || document.frm.name.value.length > 12) {
            alert("이름을 6~12자 이내로 입력해주세요.");
            document.frm.name.select();
        }
    }
</script>
<body>
    <form name="frm">
        <p>이름 : <input type="text" name="name">
        <input type="submit" value="전송" onclick="checkForm()">
    </form>
</body>
</html>
```



The screenshot shows a web form with a text input field containing the text "admin" and a "전송" (Submit) button. Below the form, an alert dialog box titled "웹 페이지 메시지" (Web Page Message) is displayed. The alert contains a warning icon and the text "이름을 6~12자 이내로 입력해주세요." (Please enter a name within 6~12 characters). There is a "확인" (OK) button at the bottom of the alert.

2. 데이터 길이 확인하기

■ [예제 8-3] 폼 페이지에 입력한 데이터(아이디와 비밀번호) 값의 길이 검사하기

JSPBook/src/main/webapp/ch08/validation03.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
```

```
02 <html>
```

```
03 <head>
```

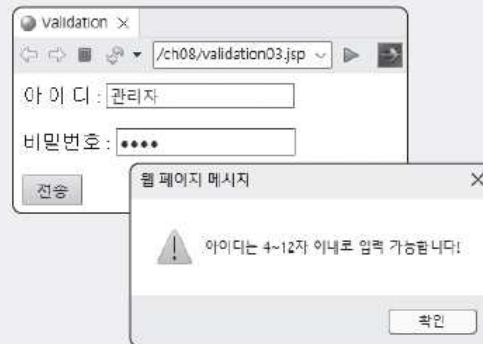
```
04 <title>Validation</title>
```

```
05 </head>
```

```
06 <script type="text/javascript">
```

```
07     function checkLogin() {  
08         var form=document.loginForm;
```

```
09  
10         if (form.id.value.length < 4 || form.id.value.length > 12) {  
11             alert("아이디는 4~12자 이내로 입력 가능합니다!");  
12             form.id.select();  
13             return;  
14         }  
15  
16         if (form.passwd.value.length < 4) {  
17             alert("비밀번호는 4자 이상으로 입력해야 합니다!");  
18             form.passwd.select();  
19             return;  
20         }  
21  
22         form.submit();  
23     }  
24 </script>
```



```
25 <body>
```

```
26 <form name="loginForm" action="validation03_process.jsp" method=  
    "post">
```

```
27     <p> 아 이 디 : <input type="text" name="id">
```

```
28     <p> 비밀번호 : <input type="password" name="passwd">
```

```
29     <p> <input type="button" value="전송" onclick="checkLogin()>
```

```
30 </form>
```

```
31 </body>
```

```
32 </html>
```

2. 데이터 길이 확인하기

■ [예제 8-3] 폼 페이지에 입력한 데이터(아이디와 비밀번호) 값의 길이 검사하기

JSPBook/src/main/webapp/ch08/validation03_process.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <title>Validation</title>
05 </head>
06 <body>
07     <h3>입력에 성공했습니다.</h3>
08     <%
09         request.setCharacterEncoding("utf-8");
10         String id=request.getParameter("id");
11         String passwd=request.getParameter("passwd");
12     %>
13     <p> 아이디 : <%= id %>
14     <p> 비밀번호 : <%= passwd %>
15 </body>
16 </html>
```



3. 숫자 여부 확인하기

■ 숫자 여부 확인하기

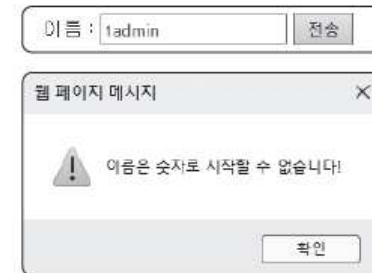
- 숫자 여부는 isNaN() 함수를 활용하여 검사
- isNaN
 - ✓ isNaN의 약자
 - ✓ isNaN() 함수의 인자 값이 숫자이면 false를 반환하고 숫자가 아니면 true를 반환

```
isNaN(document.폼이름.입력양식이름.value)
```


3. 숫자 여부 확인하기

■ 입력 데이터의 숫자 여부 검사 예

```
<%@ page contentType="text/html; charset=utf-8" %>
<html>
<head>
<title>Validation</title>
</head>
<script type="text/javascript">
    function checkForm() {
        if (!isNaN(document.frm.name.value.substr(0, 1))) {
            alert("이름은 숫자로 시작할 수 없습니다!");
            document.frm.name.select();
        }
    }
</script>
<body>
    <form name="frm">
        <p> 이름 : <input type="text" name="name">
            <input type="submit" value="전송" onclick="checkForm()">
        </form>
</body>
</html>
```



The screenshot shows a web form with a text input field containing 'tadmin' and a '전송' (Submit) button. Below the form, an alert dialog box titled '웹 페이지 메시지' (Web Page Message) is displayed. The message reads: '이름은 숫자로 시작할 수 없습니다!' (Name cannot start with a number!). There is a warning icon (exclamation mark in a triangle) and an '확인' (OK) button at the bottom of the alert box.

3. 숫자 여부 확인하기

■ [예제 8-4] 폼 페이지에 입력한 비밀번호 값이 숫자인지 검사하기

학생 > 11번줄 charAt(i)는?

교수 > https://www.w3schools.com/jsref/jsref_charat.asp

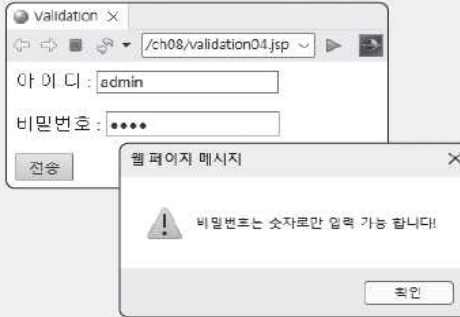
학생 > 15번줄 form.id.select()는?

교수 > https://www.w3schools.com/jsref/met_text_select.asp (Try it yourself)

Validation02.jsp의 form.id.focus() 와 유사

JSPBook/src/main/webapp/ch08/validation04.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <title>Validation</title>
05 </head>
06 <script type="text/javascript">
07     function checkLogin() {
08         var form=document.loginForm;
09
10         for (i=0; i < form.id.value.length; i++) {
11             var ch=form.id.value.charAt(i);
12
13             if ((ch<'a' || ch>'z') && (ch>'A' || ch<'Z') && (ch>'0' || ch<'9')) {
14                 alert("아이디는 영문 소문자만 입력 가능합니다!");
15                 form.id.select();
16                 return;
17             }
18         }
19     }
```



The screenshot shows a web browser window titled 'validation' with the address bar displaying '/ch08/validation04.jsp'. The login form has an '아이디' (ID) field containing 'admin' and a '비밀번호' (Password) field with masked characters. A '진송' (Submit) button is visible. An alert dialog box is displayed over the form, titled '웹 페이지 메시지' (Web Page Message), with a warning icon and the message '비밀번호는 숫자로만 입력 가능합니다!' (Password can only be entered with numbers!). The '확인' (OK) button is at the bottom right of the alert.

3. 숫자 여부 확인하기

■ [예제 8-4] 폼 페이지에 입력한 비밀번호 값이 숫자인지 검사하기

```
20     if (isNaN(form.passwd.value)) {
21         alert("비밀번호는 숫자만 입력 가능합니다!");
22         form.passwd.select();
23         return;
24     }
25
26     form.submit();
27 }
28 </script>
29 <body>
30     <form name="loginForm" action="validation04_process.jsp" method=
31         "post">
32         <p> 아 이 디 : <input type="text" name="id">
33         <p> 비밀번호 : <input type="password" name="passwd">
34         <p> <input type="button" value="전송" onclick="checkLogin()">
35     </form>
36 </body>
37 </html>
```

3. 숫자 여부 확인하기

■ [예제 8-4] 폼 페이지에 입력한 비밀번호 값이 숫자인지 검사하기

JSPBook/src/main/webapp/ch08/validation04_process.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
04 <title>Validation</title>
05 </head>
06 <body>
07     <h3>입력에 성공했습니다.</h3>
08     <%
09         request.setCharacterEncoding("utf-8");
10         String id=request.getParameter("id");
11         String passwd=request.getParameter("passwd");
12     %>
13     <p> 아이디 : <%= id %>
14     <p> 비밀번호 : <%= passwd %>
15 </body>
16 </html>
```

휴식

1. 유효성 검사의 개요
2. 기본 유효성 검사
3. 데이터 형식 유효성 검사
4. [북마켓] 도서 등록 데이터의 유효성 검사하기

Section 03

데이터 형식 유효성 검사

1. 정규 표현식 사용하기 (1학년 웹기초에서 배우지 않았음)

■ 데이터 형식 유효성 검사

- 사용자가 폼 페이지의 입력 항목에 입력한 데이터 값이 특정 형태에 적합한지 검사하기 위해 정규 표현식을 사용하는 방법

■ 정규 표현식(regular expression)

- 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 표현식
- 주민등록번호, 전화번호, 이메일과 같이 데이터 형식의 패턴이 일정한 데이터를 검사하는 데 이용함

예) `var regExpId = /^[a-zA-Zㄱ-ㅎㅏ-ㅣ가-힣]*$/;`

여기서 `^[a-zA-Zㄱ-ㅎㅏ-ㅣ가-힣]*` 는 정규표현식

`regExpId.test(id);` : id 문자열이 영문대문자와소문자, 한글자음, 한글모음, 한글로만 구성되어있는지 test. 반환값은 true 또는 false

학생> 정규표현식 없이 test하려면 비교문을 아주 많이 사용해야하는데 2줄로 해결하네요.

교수> JS 이외에도 C언어, Python등에 정규표현식 라이브러리가 있습니다.

1. 정규 표현식 사용하기

- 정규표현식 tutorial : https://www.w3schools.com/js/js_regexp.asp
- 정규표현식 reference : https://www.w3schools.com/jsref/jsref_obj_regexp.asp

<용어>

1. Modifiers
2. Groups
3. Metacharacters
4. Quantifiers
5. Methods
6. Properties

1.Modifiers : /pattern/modifiers;

ex) /w3schools/i

/w3schools/i is a regular expression.

w3schools is a pattern (to be used in a search 또는 test)

i is a modifier (modifies the search to be case-insensitive)

2.Groups : Brackets are used to find a range of characters

ex) [abc] Find any of the characters between the brackets

3.Metacharacters : characters with a special meaning

ex) \d Find a digit

4.Quantifiers : define quantities:

ex) n+ Matches any string that contains at least one n

5.Method : 정규표현식용 메소드

ex) /e/.test("The best things in life are free!");

6.Properties : 정규표현식용 Properties (reference에 사례 나옴)

1. 정규 표현식 사용하기

■ 정규 표현식 문자열을 찾는 예

regExp.exec(str) https://www.w3schools.com/jsref/jsref_regexp_exec.asp

```
<%@ page contentType="text/html; charset=utf-8" %>
<html>
<head>
<title>Validation</title>
</head>
<script type="text/javascript">
    function checkForm() {
        var regExp=/Java/i;    //var regExp=new RegExp('java','i');와 같음
        var str=document.frm.title.value;
        var result=regExp.exec(str);
        alert(result[0]);
    }
</script>
<body>
    <form name="frm">
        <p> 제목 : <input type="text" name="title">
            <input type="submit" value="전송" onclick="checkForm()">
        </form>
    </body>
</html>
```

제목 : Java Server Pages 전송



2. 정규 표현식의 표현 방법

■ 입력 데이터의 첫 글자가 문자인지 검사하는 예 (check.jsp)

■ 9번줄 의미 : `/^[a-zA-Zㄱ-ㅎ|ㅌ-ㅣ|가-힣]/` : 첫 글자가 문자인 regular expression

^ : ^n quantifier matches any string with n at the beginning of it.

https://www.w3schools.com/jsref/jsref_regexp_ncaret.asp

[a-z : [0-9] expression is used to find any character between the brackets.

https://www.w3schools.com/jsref/jsref_regexp_0-9.asp

[a-z | A-Z : Using | as OR

https://www.w3schools.com/jsref/jsref_regexp_xy.asp

■ 10번줄 의미: `regExp.test(str)`

`regExp`가 `str`에서 찾을수 있으면 `true`반환

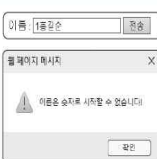
`regExp`는 첫 글자가 문자인 regular expression

https://www.w3schools.com/jsref/jsref_regexp_test.asp

예) string이 홍길순이면 `true`

string이 1홍길순이면 `false`

```
<? page contentType="text/html; charset=utf-8" %>
<html>
<head>
<title>Validation</title>
</head>
<script type="text/javascript">
    function checkForm() {
        var str=document.frm.name.value;
        var regExp = /^[a-zA-Zㄱ-ㅎ|ㅌ-ㅣ|가-힣]/;
        if (!regExp.test(str)) {
            alert("이름은 숫자로 시작할 수 없습니다!");
            return;
        }
    }
</script>
<body>
    <form name="frm">
        <p> 이름 : <input type="text" name="name">
        <input type="submit" value="전송" onclick="checkForm()">
    </form>
</body>
</html>
```



2. 정규 표현식의 표현 방법

■ [예제 8-5] 회원 가입 폼 페이지에 입력한 데이터 형식 유효성 검사하기

- 9번줄 의미 regExpId = /^[a-zA-Zㄱ-ㅎㅌ-ㅣ가-힣]/

id 첫 글자가 문자인 regular expression

- 10번줄 의미 regExpName = /^[가-힣]*\$/;

name 첫글자가 한글이고 * (zero or more반복) \$ (끝글자가 한글)

- 11번줄 의미 regExpPasswd = /^[0-9]*\$/;

passwd 첫글자가 숫자이고 * (zero or more반복) \$ (끝글자가 숫자)

- 12번줄 의미 regExpPhone = /^[Wd{3}-Wd{3,4}-Wd{4}]\$/;

phone 전화번호 형태인지 검사

Wd metacharacter matches digits from 0 to 9.

n{X} quantifier matches any string that contains a sequence of X n's

n{X,Y} quantifier matches any string that contains a sequence of X to Y n's.

phone이 첫글자가 숫자이고 숫자가3회-숫자가3에서4회-숫자가4회인 정규표현식인지 검사

예) 031-123-4567

- 13번줄의 의미 regExpEmail = /^[0-9a-zA-Z]([-_W.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_W.]?[0-9a-zA-Z])*W[a-zA-Z]{2,3}\$/i;

email 이메일형태인지 검사

^n quantifier matches any string with n at the beginning of it.

[0-9a-zA-Z] Brackets [abc] specifies matches for the characters inside the brackets.

[-_W.] -또는 _ 또는 . (.만 있으면 metacharacter로 인식)

? n? quantifier matches any string that contains zero or one occurrences of n.

[0-9a-zA-Z] Brackets [abc] specifies matches for the characters inside the brackets.

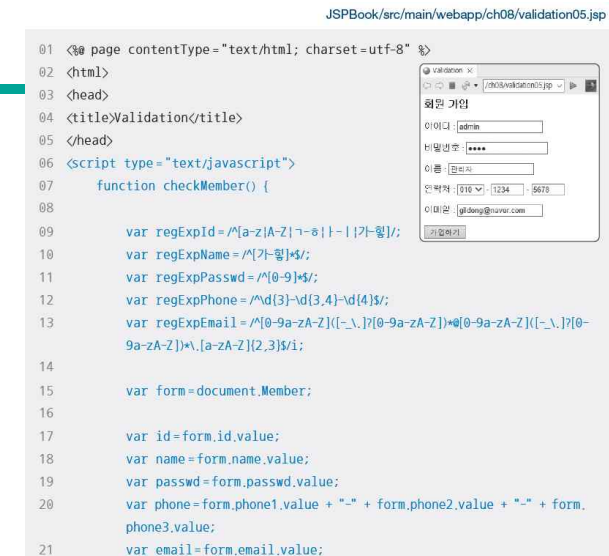
n* quantifier matches any string that contains zero or more occurrences of n.

@

[0-9a-zA-Z]([-_W.]?[0-9a-zA-Z])* : @앞부분과 동일설명

[a-zA-Z]{2,3} : n{X,Y} quantifier matches any string that contains a sequence of X to Y n's.

[a-zA-Z]{2,3}\$: n\$ quantifier matches any string with n at the end of it 영문자로 종료



2. 정규 표현식의 표현 방법

■ [예제 8-5] 회원 가입 폼 페이지에 입력한 데이터 형식 유효성 검사하기

학생> 교수님~ regExpEmail 너무 복잡해요

정규 표현식을 바꾸어 보기 쉽고 입력값도 쉽게 바꾸어 보고 싶어요

교수> 9주차 추가코드에서 match_change.jsp 사용하여 검사

■ regExpPasswd = /^[0-9]*\$/

passwd 첫글자가 숫자이고 * (zero or more반복) \$ (끝글자가 숫자)

Match : 0 , 01

Unmatch : a9, 9a

■ regExpEmail = /^[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*\.[a-zA-Z]{2,3}\$/i;

email 이메일형태인지 검사

Match : 0.a@0.a.ab

Unmatch : 0.a

2. 정규 표현식의 표현 방법

■ [예제 8-5] 회원 가입 폼 페이지에 입력한 데이터 형식 유효성 검사하기

```
22
23     if (!RegExpId.test(id)) {
24         alert("아이디는 문자로 시작해주세요!");
25         form.id.select();
26         return;
27     }
28     if (!RegExpName.test(name)) {
29         alert("이름은 한글만 입력해주세요!");
30         return;
31     }
32     if (!RegExpPasswd.test(passwd)) {
33         alert("비밀번호는 숫자만 입력해주세요!");
34         return;
35     }
36     if (!RegExpPhone.test(phone)) {
37         alert("연락처 입력을 확인해주세요!");
38         return;
39     }
40     if (!RegExpEmail.test(email)) {
41         alert("이메일 입력을 확인해주세요!");
42         return;
43     }
44
45     form.submit();
46
47 }
48 </script>
```

2. 정규 표현식의 표현 방법

■ [예제 8-5] 회원 가입 폼 페이지에 입력한 데이터 형식 유효성 검사하기

```
49 <body>
50     <h3>회원 가입</h3>
51     <form action="validation05_process.jsp" name="Member" method="post">
52         <p>아이디 : <input type="text" name="id">
53         <p>비밀번호 : <input type="password" name="passwd">
54         <p>이름 : <input type="text" name="name">
55         <p>연락처 : <select name="phone1">
56             <option value="010">010</option>
57             <option value="011">011</option>
58             <option value="016">016</option>
59             <option value="017">017</option>
60             <option value="019">019</option>
61         </select> - <input type="text" maxlength="4" size="4" name
            ="phone2"> -
62         <input type="text" maxlength="4" size="4" name="phone3">
63         <p>이메일 : <input type="text" name="email">
64         <p><input type="button" value="가입하기" onclick="checkMember()>
65     </form>
66 </body>
67 </html>
```

2. 정규 표현식의 표현 방법

■ [예제 8-5] 회원 가입 폼 페이지에 입력한 데이터 형식 유효성 검사하기

JSPBook/src/main/webapp/ch08/validation05_process.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <title>Validation</title>
05 </head>
06 <body>
07     <%
08         request.setCharacterEncoding("UTF-8");
09     %>
10     <p> 아이디 : <%=request.getParameter("id") %>
11     <p> 비밀번호 : <%=request.getParameter("passwd") %>
12     <p> 이름 : <%=request.getParameter("name") %>
13     <p> 연락처 : <%=request.getParameter("phone1") %>-<%=request.getParameter
14                 ("phone2") %>-<%=request.getParameter("phone3") %>
15 </body>
16 </html>
```



3. 웹 프로그래밍 언어의 종류와 특징

학생> 다양한 정규표현식 보고싶어요

교수> <https://regexlib.com> > Browse Expresssions > Email > Overly Simple Match와 Unmatch 문자열을 match_change.jsp 사용하여 검사

NOTE 자주 사용되는 패턴

다음은 폼 페이지에 입력 양식의 데이터 형식으로 주로 사용되는 정규 표현식의 예입니다.

데이터 형식	패턴
숫자만	^[0-9]*\$
영문자만	^[a-zA-Z]*\$
한글만	^[가-]*\$
영문자와 숫자만	^[a-zA-Z0-9]*\$
이메일	^[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_\.]?[0-9a-zA-Z])*\.[a-zA-Z]{2,3}\$/i
휴대전화	^01(?:0 1 [6-9]) - (?:\d{3} \d{4}) - \d{4}\$ ^\d{3}-\d{3,4}-\d{4}\$
전화번호	^\d{2,3} - \d{3,4} - \d{4}\$
주민등록번호	\d{6} \- [1-4]\d{6}
IP 주소	([0-9]{1,3}) \. ([0-9]{1,3}) \. ([0-9]{1,3}) \. ([0-9]{1,3})
URL	^(file gopher news nnntp telnet https? ftp? sftp):\\\/\\\/([a-z0-9-]+\.)+[a-z0-9]{2,4}.*\$
날짜	\d{1,2}\/\d{1,2}\/\d{2,4}\$

휴식

1. 유효성 검사의 개요
2. 기본 유효성 검사
3. 데이터 형식 유효성 검사
4. [북마켓] 도서 등록 데이터의 유효성 검사하기

Section 04

[북마켓] 도서 등록 데이터의
유효성 검사하기

[북마켓] 도서 등록 데이터의 유효성 검사하기

■ 프로젝트 로드맵

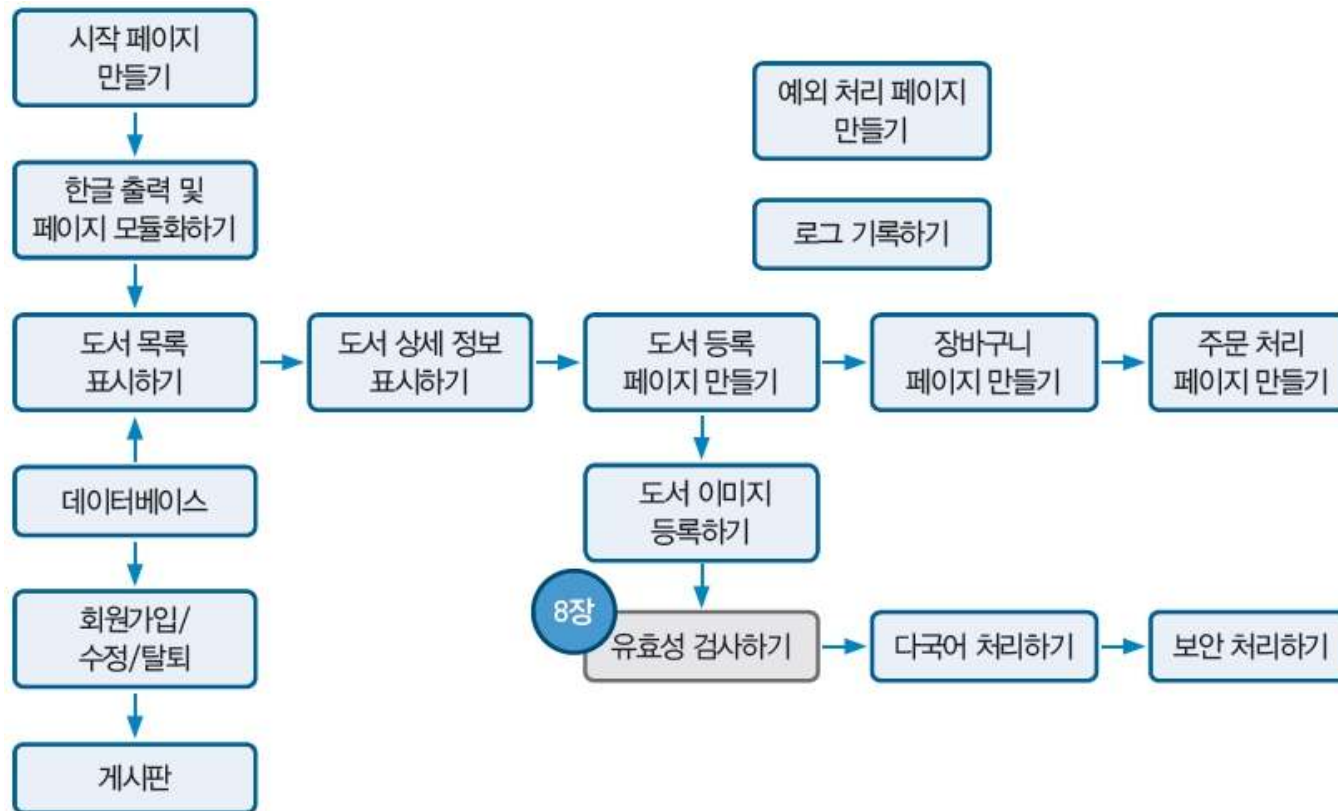


그림 8-1 도서 등록 데이터의 유효성 검사하기

■ 프로젝트 실행 결과 (addBook.jsp > 오버튼)

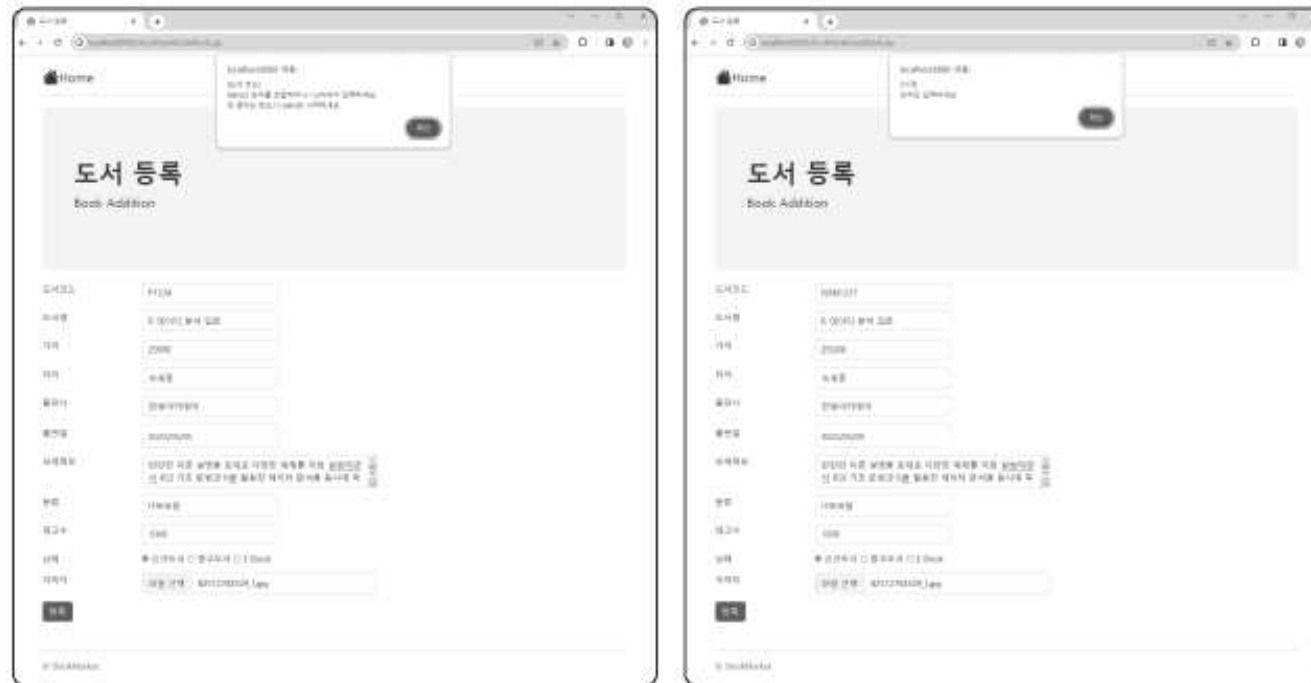


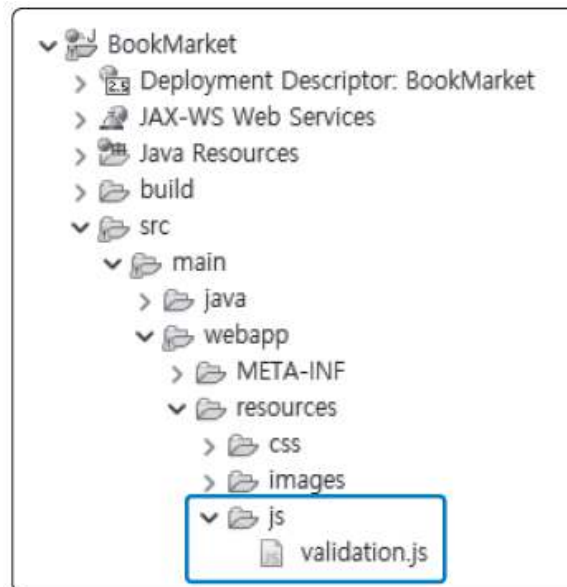
그림 8-2 완성된 북마켓 도서 등록 데이터의 유효성 검사

[북마켓] 도서 등록 페이지 만들기

■ [예제 8-6] 도서 등록 페이지의 유효성 검사하기

1. 자바스크립트 관리 폴더 만들기

- /src/main/webapp/resources/ 폴더에 js 폴더 생성하고 validation.js 생성하기



[북마켓] 도서 등록 페이지 만들기

■ [예제 8-6] 도서 등록 페이지의 유효성 검사하기

2. 유효성 검사를 위한 핸들러 함수 작성하기

BookMarket/src/main/webapp/resources/js/validation.js

```
01 function CheckAddBook() {
02
03     var bookId=document.getElementById("bookId");
04     var name=document.getElementById("name");
05     var unitPrice=document.getElementById("unitPrice");
06     var unitsInStock=document.getElementById("unitsInStock");
07     var description=document.getElementById("description");
08
09     // 도서아이디 체크
10     if (!check(/^[ISBN[0-9]{4,11}]$/, bookId, "[도서 코드]\nISBN과 숫자를 조합하여 5~12
        자까지 입력하세요\n첫 글자는 반드시 ISBN로 시작하세요"))
11         return false;
12
13     // 도서명 체크
14     if (name.value.length < 4 || name.value.length > 50) {
15         alert("[도서명]\n최소 4자에서 최대 50자까지 입력하세요");
16         name.focus();
17         return false;
18     }
19 }
```

[북마켓] 도서 등록 페이지 만들기

■ [예제 8-6] 도서 등록 페이지의 유효성 검사하기

```
20 // 도서 가격 체크
21 if (unitPrice.value.length==0 || isNaN(unitPrice.value)) {
22     alert("[가격]\n숫자만 입력하세요");
23     unitPrice.focus();
24     return false;
25 }
26
27 if (unitPrice.value < 0) {
28     alert("[가격]\n음수를 입력할 수 없습니다");
29     unitPrice.focus();
30     return false;
31 }
32
33 // 재고 수 체크
34 if (isNaN(unitsInStock.value)) {
35     alert("[재고 수]\n숫자만 입력하세요");
36     unitsInStock.focus();
37     return false;
38 }
39
40 if (description.value.length < 100) {
41     alert("[상세설명]\n최소 100자 이상 입력하세요");
42     description.focus();
43     return false;
44 }
45
```

[북마켓] 도서 등록 페이지 만들기

■ [예제 8-6] 도서 등록 페이지의 유효성 검사하기

```
46  function check(regExp, e, msg) {  
47      if (regExp.test(e.value)) {  
48          return true;  
49      }  
50      alert(msg);  
51      e.focus();  
52      return false;  
53  }  
54  
55  document.newBook.submit()  
56 }
```


[북마켓] 도서 등록 페이지 만들기

■ [예제 8-6] 도서 등록 페이지의 유효성 검사하기

3. 도서 등록 페이지 수정하기

BookMarket/src/main/webapp/addBook.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <link rel="stylesheet" href="./resources/css/bootstrap.min.css" />
05 <script type="text/javascript" src="./resources/js/validation.js"></script>
06 <title>도서 등록</title>
07 </head>
08 <body>
09 ... (생략) ...
10 <div class="mb-3 row">
11 <label class="col-sm-2">도서 코드</label>
12 <div class="col-sm-3">
13 <input type="text" id="bookId" name="bookId" class="form-control">
14 </div>
15 </div>
16 <div class="mb-3 row">
17 <label class="col-sm-2">도서명</label>
18 <div class="col-sm-3">
19 <input type="text" id="name" name="name" class="form-control">
20 </div>
21 </div>
```

[북마켓] 도서 등록 페이지 만들기

■ [예제 8-6] 도서 등록 페이지의 유효성 검사하기

```
22     <div class="mb-3 row">
23         <label class="col-sm-2">가격</label>
24         <div class="col-sm-3">
25             <input type="text" id="unitPrice" name="unitPrice" class="
                "form-control">
26         </div>
27     </div>
28     ...<생략>...
29     <div class="mb-3 row">
30         <label class="col-sm-2">상세정보</label>
31         <div class="col-sm-5">
32             <textarea name="description" id="description" cols="50" rows="2"
                class="form-control" placeholder="100자 이상 적어주세요"></textarea>
33         </div>
34     </div>
35     ...<생략>...
36     <div class="mb-3 row">
37         <label class="col-sm-2">재고 수</label>
38         <div class="col-sm-3">
39             <input type="text" id="unitsInStock" name="unitsInStock" class
                ="form-control">
40         </div>
41     </div>
42     ...<생략>...
```

[북마켓] 도서 등록 페이지 만들기

■ [예제 8-6] 도서 등록 페이지의 유효성 검사하기

```
43     <div class="mb-3 row">
44         <div class="col-sm-offset-2 col-sm-10">
45             <input type="button" class="btn btn-primary" value="등록" onclick
              = "CheckAddBook()">
46         </div>
47     </div>
48     ... (생략) ...
```

[북마켓] 도서 등록 페이지 만들기

학생> addBook.jsp로 시작하고 addBook.jsp의 5번줄에서 validation.js를 포함하므로 크롬에서 validation.js가 수행됩니다. Validation.js 내용을 보니 5개 항목 bookId, name, unitPrice, unitsInStock, description 에 대해 유효성 검사를 합니다.

검사 내용을 보니 JSPBook의 validation01.jsp~validation05.jsp 내용이라 어렵지 않습니다. F12 디버깅하고 싶습니다.

교수> JS는 크롬에서 실행되므로 디버깅은 eclipse가 아닌 F12로 디버깅합니다.

1) 북마켓 addBook.jsp 실행

학생> validation.js파일은 어디에 있나요?

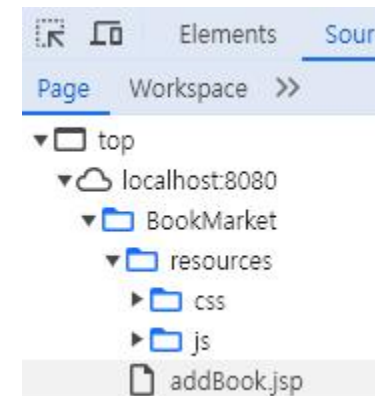
2) 크롬에서 validation.js의 64번줄에 중단점설정

3) bookId, name, unitPrice, unitsInStock, description 유효한값 입력
bookId: ISBN1234 name: 12345 unitPrice: 1 unitsInStock: 1

4) 중단점도착하면 Local창에서 name.value변수 확인

5) 중단점해제

6) Resume 버튼



학생> Local창 name.value에서 value가 안보입니다.

교수> (...)을 클릭합니다

```
validationMessage: ""
▶ validity: ValidityState {
  value: "12345"
  valueAsDate: null
  ...
}
```

학생> 에러나는 데요

교수> 에러메시지 보고 processAddBook.jsp 13번줄의 realFolder위치를
7장 북마켓처럼 변경

```
onsecuritypolicy
onseeked: null
(...)
inert: false
innerHTML: "<div></div>"
innerText: "OK OK"
```

이제 나는 할 수 있다

1. 나는 유효성 검사의 개념을 이해한다.
2. 나는 유효성 검사를 위한 핸들러 함수의 작성 방법을 익힌다.
3. 나는 유효성 검사를 처리하는 방법을 익힌다.
4. 나는 유효성 검사를 이용하여 도서 상품 등록 데이터를 검증한다.

1층(1장): 쇼핑몰 프로젝트 생성하기

7층(7장): 도서이미지 등록하기

8층(8장): 도서등록 데이터의 유효성 검사하기

* 정답이 있는 연습문제 꼭 해결해 보세요.