

8 유효성 검사

1

유효성 검사의 개요

2

기본 유효성 검사

3

데이터 형식 유효성 검사

4

[웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기

학습목표

- 유효성 검사의 개념을 이해합니다.
- 유효성 검사를 위한 핸들러 함수의 작성 방법을 익힙니다.
- 유효성 검사를 처리하는 방법을 익힙니다.
- 유효성 검사를 이용하여 상품 등록 데이터를 검증합니다.

1. 유효성 검사의 개요

❖ 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
 - 폼 페이지에서 나이를 입력할 때 숫자를 인식하는 검사, 회원 가입 시 아이디 중복 검사, 로그인 인증 시 아이디와 비밀번호 검사, IP 패킷 검사 등

NOTE_ 유효성 검사가 필요한 이유와 검사 항목

웹 애플리케이션에서 폼 데이터의 유효성 검사가 필요한 가장 큰 이유는 보안 공격, 잘못된 데이터, 사용자의 실수로 예상 가능한 오류 등을 방지할 수 있기 때문입니다. 유효성 검사 기능을 이용하면 다양한 사용자가 폼 데이터를 입력해도 폼 페이지의 동일한 입력 양식과 형태를 유지할 수 있습니다.

입력 양식이 있는 폼 페이지를 만들 때 반드시 점검해야 할 유효성 검사의 항목은 다음과 같습니다.

- 입력 데이터가 null인지 확인하는 유효성 검사
- 날짜나 이메일을 입력할 때 형식에 맞는지 확인하는 유효성 검사
- 나이를 입력할 때 숫자인지 확인하는 유효성 검사
- 입력 데이터의 제한 길이를 초과했는지 확인하는 유효성 검사
- 로그인 인증 시 아이디와 비밀번호를 확인하는 유효성 검사
- 회원 가입 시 아이디 중복 여부를 확인하는 유효성 검사

1. 유효성 검사의 개요

❖ 유효성 검사를 위한 핸들러 함수

- 핸들러 함수는 폼 페이지에서 이벤트가 발생했을 때(<submit>를 클릭한 경우)의 유효성 검사를 위해 매핑하는 메소드
- 자바스크립트를 이용하여 유효성 검사를 위한 코드를 작성
 - 자바스크립트는 웹 브라우저에서 유효성 검사를 처리하므로 서버에서 처리하는 것보다 속도가 빠르고 서버에 과부하를 주지 않음
- 사용자가 폼 페이지에 입력한 데이터 값이 서버로 전송되기 전에 특정 규칙에 맞게 입력되었는지를 검사
- 입력된 데이터가 유효성 검사를 통과하면 서버로 전송하고, 그렇지 않으면 서버 전송을 취소하고 사용자에게 오류 메시지를 보여주는 역할을 함

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 상관없음.
- ❸ 폼 페이지에서 입력된 데이터 값을 핸들러 함수로 가져오기 위해 form 태그의 name 속성 또는 forms 객체를 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

```
<script type="text/javascript">
    function 핸들러 함수(){
        var str = document.폼 이름.입력항목 이름.value;
    }
</script>

<form name="폼 이름">
    ...(생략)...
    <input type="submit" onclick="핸들러 함수()">
</form>
```

1. 유효성 검사의 개요

```
<? 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">
```

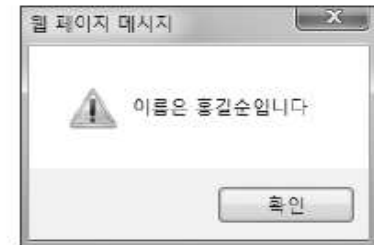
```
        <input type="submit" value="전송" onclick="checkForm()">
```

```
    </form>
```

```
</body>
```

```
</html>
```

이름 : 홍길순

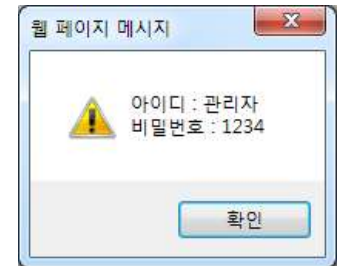
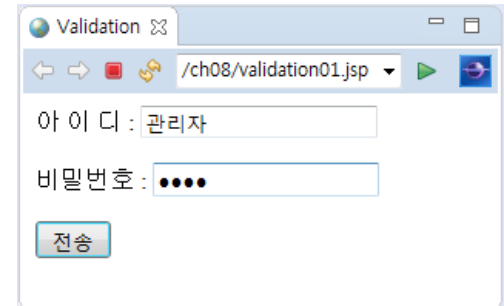


1. 유효성 검사의 개요

예제 8-1 폼 페이지에 입력한 아이디와 비밀번호 출력하기

JSPBook/WebContent/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>
```



1. 유효성 검사의 개요

❖ 유효성 검사 처리 방법

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

2. 기본 유효성 검사

❖ 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

2. 기본 유효성 검사

❖ 데이터 유무 확인하기

- 데이터 값의 유무에 대한 검사
 - 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.폼 이름.입력양식 이름.value==" "
```

2. 기본 유효성 검사

[입력 데이터의 유무 검사 예]

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

```
<html>
```

```
<head>
```

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

```
</head>
```

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

```
    }
```

```
</script>
```

```
<body>
```

```
    <form name="frm">
```

```
        <p> 이름 : <input type="text" name="name">
```

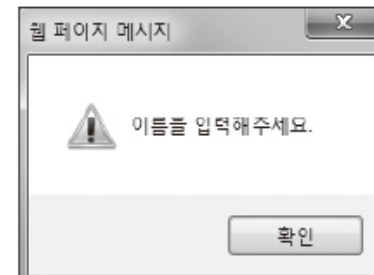
```
        <input type="submit" value="전송" onclick="checkForm()">
```

```
    </form>
```

```
</body>
```

```
</html>
```

이름 :



2. 기본 유효성 검사

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

JSPBook/WebContent/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>
```

Validation

/ch08/validation02.jsp

아 이 디 : 관리자

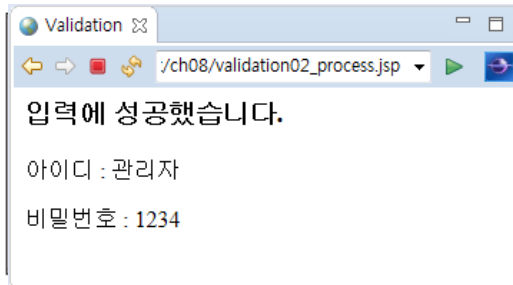
비밀번호 :

전송

2. 기본 유효성 검사

JSPBook/WebContent/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>
```



2. 기본 유효성 검사

❖ 데이터 길이 확인하기

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

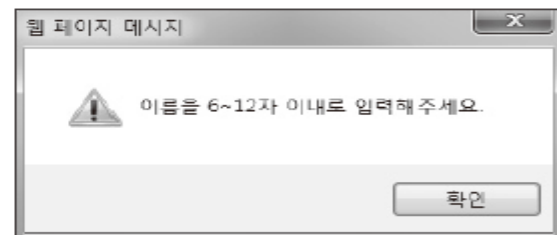
```
document.폼 이름.입력양식 이름.value.length
```

2. 기본 유효성 검사

[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>
```

이름 :

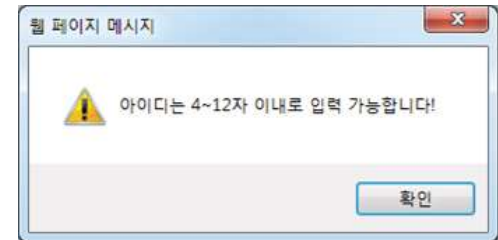


2. 기본 유효성 검사

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

JSPBook/WebContent/ch08/validation03.jsp

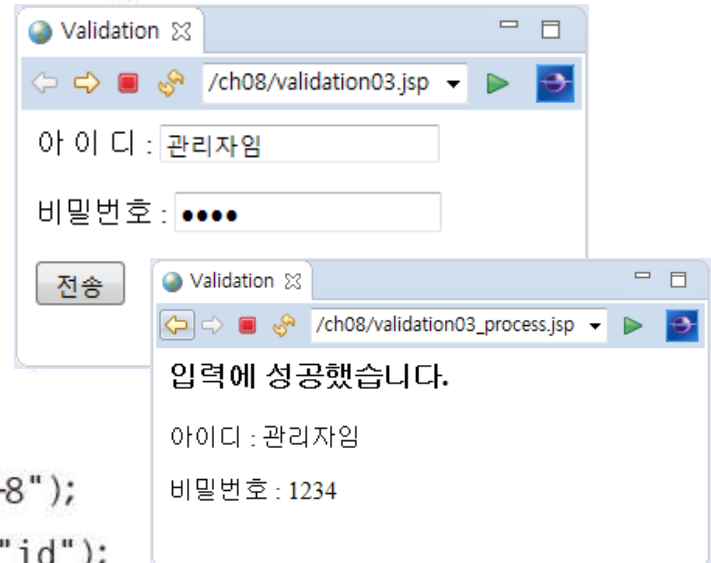
```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Validation</title>
05 </head>
06 <script type="text/javascript">
07     function checkLogin(){
08         var form = document.loginForm;
09
10         if (form.id.value.length < 4 || form.id.value.length > 12){
11             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. 기본 유효성 검사

JSPBook/WebContent/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>
```



2. 기본 유효성 검사

❖ 숫자 여부 확인하기

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

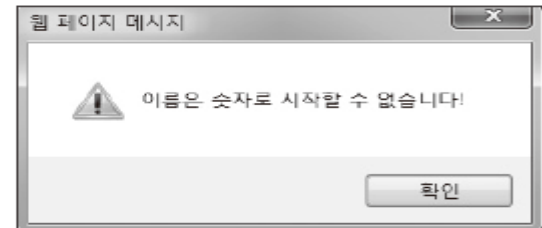
```
isNaN(document.폼 이름.입력양식 이름.value)
```

2. 기본 유효성 검사

[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>
```

이름 :

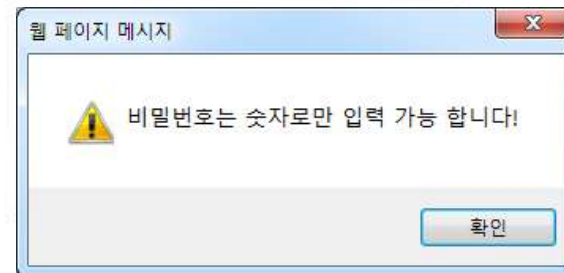


2. 기본 유효성 검사

예제 8-4 폼 페이지에 입력한 비밀번호 값이 숫자인지 검사하기

JSPBook/WebContent/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     }
```



2. 기본 유효성 검사

```
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="post">
31         <p> 아 이 디 : <input type="text" name="id">
32         <p> 비밀번호 : <input type="password" name="passwd">
33         <p> <input type="button" value="전송" onclick="checkLogin()"/>
34     </form>
35 </body>
36 <html>
```

2. 기본 유효성 검사

JSPBook/WebContent/ch08/validation04_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. 데이터 형식 유효성 검사

❖ 데이터 형식 유효성 검사

- 사용자가 폼 페이지의 입력 항목에 입력한 데이터 값이 특정 형태에 적합한지 검사하기 위해 정규 표현식(regular expression)을 사용하는 방법
- 기본 유효성검사보다 복잡

3. 데이터 형식 유효성 검사

❖ 정규 표현식 사용하기

■ 정규 표현식

- 특정한 규칙을 가진 문자열의 집합을 표현하는 데 사용하는 형식 언어
- 문자열의 특정 형태를 찾아내기 위해 패턴으로 표현한 수식
- 주민등록번호, 전화번호, 이메일과 같이 데이터 형식의 패턴이 일정한 데이터를 검사하는 데 이용

■ 정규 표현식의 사용 형식

- 객체 초기화(object initializer)를 사용하는 방법으로, 입력된 표현식이 거의 바뀌지 않는 상수 형태일 때 주로 사용

```
var 변수 이름 = /정규 표현식/[Flag];
```

Flag	설명
i	Ignore Case: 문자열의 대문자와 소문자를 구별하지 않고 검출합니다.
g	Global: 문자열 내의 모든 패턴을 검출합니다.
m	Multi Line: 문자열에 줄 바꿈 행이 있는지 검출합니다.

- RegExp 객체를 이용하는 방법으로, 정규 표현식이 자주 변경될 때 주로 사용

```
var 변수 이름 = new RegExp('정규 표현식', ['Flag']);
```


3. 데이터 형식 유효성 검사

❖ 정규 표현식의 메소드 종류

메소드	설명
test()	매개변수 값으로 전달되는 문자열이 정규 표현식에 부합한지 판단하여 true/false를 반환합니다.
exec()	매개변수 값으로 전달되는 문자열에서 정규 표현식에 부합된 문자열을 추출하여 반환합니다.

```
<%@ 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 같다
        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



3. 데이터 형식 유효성 검사

❖ 정규 표현식의 표현 방법

■ 기본 메타 문자의 종류

메타 문자	설명
<code>^x</code>	문자열이 x로 시작됩니다.
<code>x\$</code>	문자열이 x로 종료됩니다.
<code>x</code>	임의의 한 문자를 표현합니다(문자열이 x로 끝남).
<code>x+</code>	x가 한 번 이상 반복됩니다.
<code>x?</code>	x가 존재하거나 존재하지 않습니다.
<code>x*</code>	x가 0번 이상 반복됩니다.
<code>x y</code>	x 또는 y를 찾습니다(or 연산자를 의미).
<code>(x)</code>	() 안의 내용을 캡처하고 그룹화합니다.
<code>(x)(y)</code>	그룹화할 때 자동으로 앞에서부터 그룹 번호를 부여해서 캡처합니다. 그룹화된 결과 데이터는 배열 형식으로 들어갑니다.
<code>(x)(?y)</code>	캡처하지 않은 그룹을 생성할 경우 ?를 사용합니다. 결과 값 배열에 캡처하지 않은 그룹은 들어가지 않습니다.
<code>x{n}</code>	x를 n번 반복한 문자를 찾습니다.
<code>x{n,}</code>	x를 n번 이상 반복한 문자를 찾습니다.
<code>x{n,m}</code>	x를 n번 이상 m번 이하 반복한 문자를 찾습니다.

3. 데이터 형식 유효성 검사

❖ 정규 표현식의 표현 방법

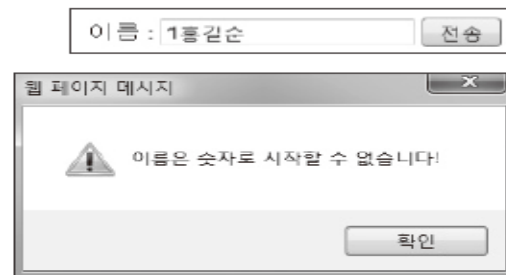
■ 문자 클래스의 종류

문자 클래스	설명
[xy]	x 또는 y를 찾습니다.
[^xy]	x, y를 제외하고 문자 하나를 찾습니다(문자 클래스 내의 ^은 not을 의미).
[x-z]	x부터 z 사이의 문자 중 하나를 찾습니다.
w^	^(특수문자)을 식에 문자 자체로 포함합니다.
wb	문자와 공백 사이의 문자를 찾습니다.
wB	공백을 제외한 문자와 문자 사이의 문자를 찾습니다.
wd	숫자를 찾습니다.
WD	숫자가 아닌 값을 찾습니다.
ws	공백 문자를 찾습니다.
WS	공백이 아닌 문자를 찾습니다.

문자 클래스	설명
wt	Tab 문자를 찾습니다.
wv	Vertical Tab 문자를 찾습니다.
ww	알파벳 + 숫자 + _을 찾습니다.
WW	알파벳 + 숫자 + _을 제외한 모든 문자를 찾습니다.

3. 데이터 형식 유효성 검사

```
<? 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-Z|ㄱ-ㅎ|ㅏ-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>
```



3. 데이터 형식 유효성 검사

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

JSPBook/WebContent/ch08/validation05.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 checkMember() {
08
09         var regExpId = /^[a-zA-Z|ㄱ-ㅎ|ㅏ-ㅣ|가-힣]/;
10         var regExpName = /^[가-힣]*$/;
11         var regExpPasswd = /^[0-9]*$/;
12         var regExpPhone = /^\d{3}-\d{3,4}-\d{4}$/;
13         var regExpEmail = /^[0-9a-zA-Z]([-_.]?[0-9a-zA-Z])*@[0-9a-zA-Z]
14         ([-_.]?[0-9a-zA-Z])*\. [a-zA-Z]{2,3}$/i;
15
16
17         var form = document.Member;
18
19         var id = form.id.value;
20         var name = form.name.value;
21         var passwd = form.passwd.value;
22         var phone = form.phone1.value + "-" + form.phone2.value + "-" + form.
23         phone3.value;
```

Validation

/ch08/validation05.jsp

회원 가입

아이디 : admin

비밀번호 :

이름 : 관리자

연락처 : 010 - 9002 - 7370

이메일 : songsnail@naver.com

가입하기

3. 데이터 형식 유효성 검사

```
21     var email = form.email.value;
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>
```

3. 데이터 형식 유효성 검사

```
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>
```

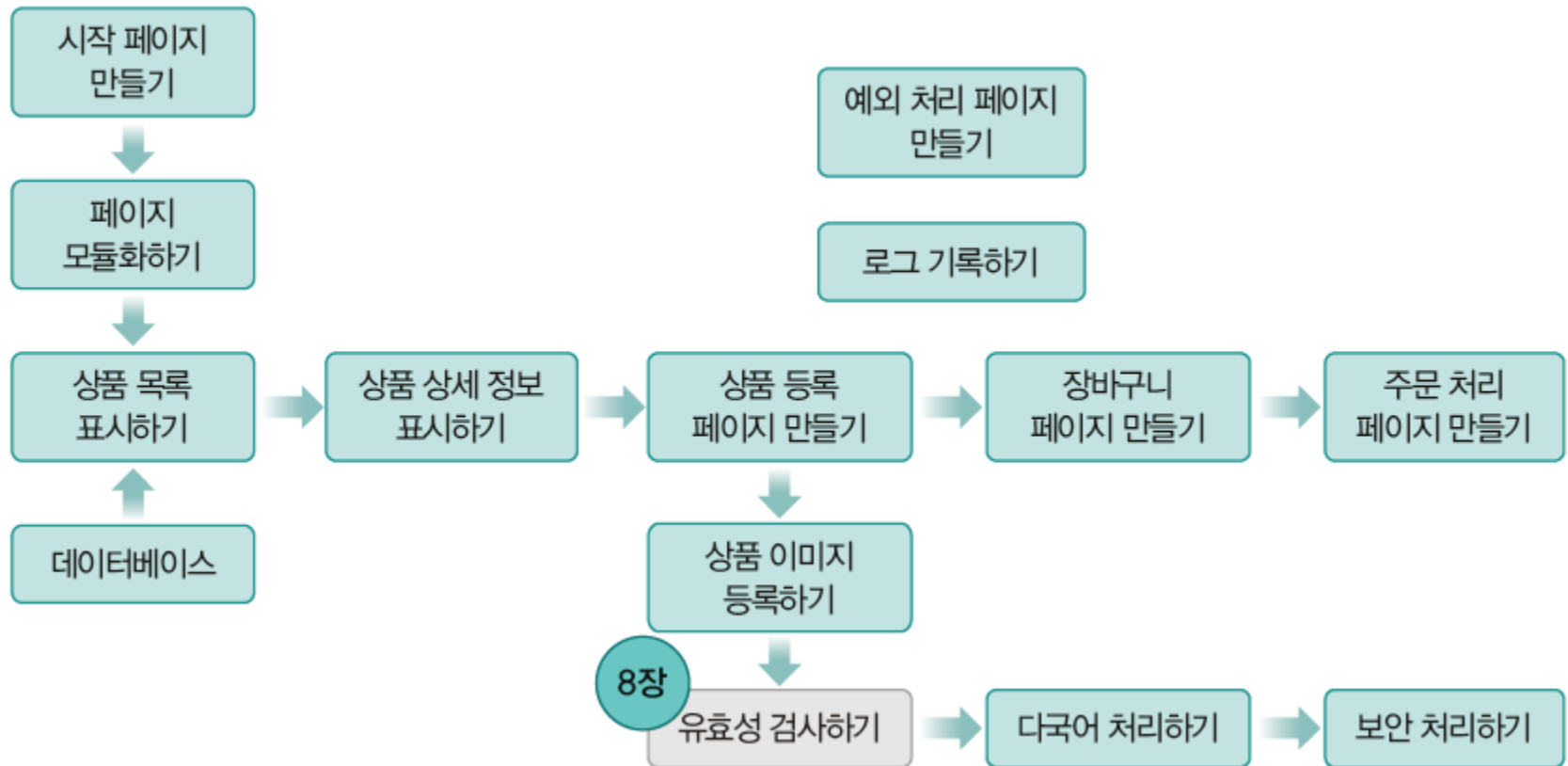
3. 데이터 형식 유효성 검사

JSPBook/WebContent/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>
```



4. [웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기



4. [웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기

The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/WebMarket/addProduct.jsp`. The page has a dark header with the word "Home" and a large light gray section with the title "상품 등록" (Product Registration). Below this is a form with the following fields and values:

Field	Value
상품코드 (Product Code)	1234
상품명 (Product Name)	Galaxy Note5
가격 (Price)	900000
상세정보 (Detailed Information)	212 8*126 6*6 6mm, Super AMOLED display, Octa-Core processor
제조사 (Manufacturer)	삼성 (Samsung)
분류 (Category)	핸드폰 (Handphone)
재고수 (Inventory)	100
상태 (Status)	<input checked="" type="radio"/> 신규제품 (New Product) <input type="radio"/> 중고제품 (Used Product) <input type="radio"/> 재생제품 (Refurbished Product)
이미지 (Image)	C:\P1237.jpg

At the bottom left of the form is a blue button labeled "등록" (Register). A modal dialog box titled "웹 페이지 메시지" (Web Page Message) is overlaid on the right side of the form. It contains a yellow warning icon and the following text:

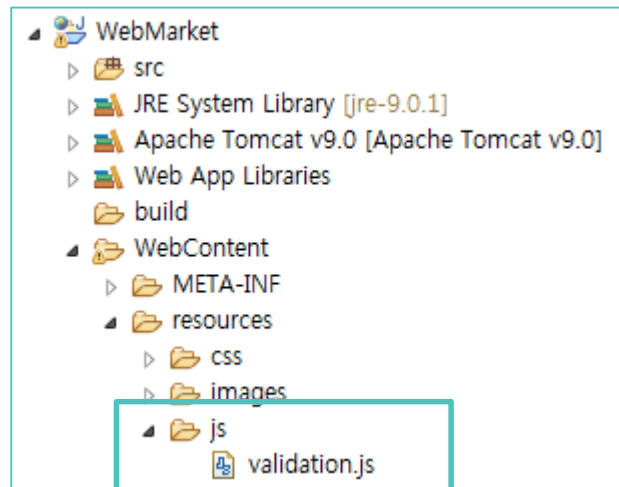
[상품코드]
P와 숫자를 조합하여 5~12자까지 입력하세요
첫글자는 반드시 P로 시작하세요

At the bottom right of the dialog is a button labeled "확인" (Confirm).

4. [웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기

예제 8-6 상품 등록 페이지의 유효성 검사하기

- 자바스크립트 관리 폴더 만들기



4. [웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기

■ 유효성 검사를 위한 핸들러 함수 작성하기

WebMarket/WebContent/resources/js/validation.js

```
01 function CheckAddProduct() {
02
03     var productId = document.getElementById("productId");
04     var name = document.getElementById("name");
05     var unitPrice = document.getElementById("unitPrice");
06     var unitsInStock = document.getElementById("unitsInStock");
07
08     //상품 아이디 체크
09     if (!check(/^P[0-9]{4,11}$/, productId,
10         "[상품 코드]\nP와 숫자를 조합하여 5~12자까지 입력하세요\n첫 글자는 반드시 P로 시작하세요"))
11         return false;
12     //상품명 체크
13     if (name.value.length < 4 || name.value.length > 12) {
14         alert("[상품명]\n최소 4자에서 최대 12자까지 입력하세요");
15         name.select();
16         name.focus();
17         return false;
18     }
19     //상품 가격 체크
20     if (unitPrice.value.length == 0 || isNaN(unitPrice.value)) {
21         alert("[가격]\n숫자만 입력하세요");
```

4. [웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기

```
22     unitPrice.select();
23     unitPrice.focus();
24     return false;
25 }
26
27 if (unitPrice.value < 0) {
28     alert("[가격]\n음수는 입력할 수 없습니다");
29     unitPrice.select();
30     unitPrice.focus();
31     return false;
32 } else if (!check(/^\d+(?:[.]\d)?\d$/, unitPrice,
33     "[가격]\n소수점 둘째 자리까지만 입력하세요"))
34     return false;
35
36 //재고 수 체크
37 if (isNaN(unitsInStock.value)) {
38     alert("[재고 수]\n숫자만 입력하세요");
39     unitsInStock.select();
40     unitsInStock.focus();
41     return false;
42 }
43
44 function check(regExp, e, msg) {
45
46     if (regExp.test(e.value)) {
47         return true;
48     }
49     alert(msg);
50     e.select();
51     e.focus();
52     return false;
53 }
54
55 document.newProduct.submit()
56 }
```

```
var productId = document.newProduct.productId.value;
var name = document.newProduct.name.value;
var unitPrice = document.newProduct.unitPrice.value;
var unitsInStock = document.newProduct.unitsInStock.value;
```

4. [웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기

■ 상품 등록 페이지 수정하기

WebMarket/WebContent/addProduct.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>상품 등록</title>
07 </head>
08 <body>
09 ... (생략) ...
10         <div class="form-group row">
11             <label class="col-sm-2">상품 코드</label>
12             <div class="col-sm-3">
13                 <input type="text" id="productId" name="productId"
14                     class="form-control">
15             </div>
16         </div>
17         <div class="form-group row">
18             <label class="col-sm-2">상품명</label>
19             <div class="col-sm-3">
20                 <input type="text" id="name" name="name" class="form-
21                     control">
22             </div>
23         </div>
24     </div>
25 </body>
26 </html>
```

4. [웹 쇼핑몰] 상품 등록 데이터의 유효성 검사하기

```

22     <div class="form-group row">
23         <label class="col-sm-2">가격</label>
24         <div class="col-sm-3">
25             <input type="text" id="unitPrice" name="unitPrice"
26                 class="form-control">
27         </div>
28     </div>
29     ... (생략) ...
30     <div class="form-group row">
31         <label class="col-sm-2">재고 수</label>
32         <div class="col-sm-3">
33             <input type="text" id="unitsInStock" name=
34                 "unitsInStock" class="form-control">
35         </div>
36     </div>
37     ... (생략) ...
38     <div class="form-group row">
39         <div class="col-sm-offset-2 col-sm-10">
40             <input type="button" class="btn btn-primary" value=
41                 "등록" onclick="CheckAddProduct()">
42         </div>
43     </div>
44     ... (생략) ...

```

