

6장. 유효성 검사

Validation / Regular Expression



유효성 검사(validation)

유효성 검사(validation)

- 사용자가 폼 페이지에서 입력한 데이터 값이 서버로 전송되기 전에 특정 규칙에 맞게 입력되었는지 검증하는 것을 말한다.
예를 들면, 회원 가입시 아이디나 비밀번호, 연락처등 검사, 아이디 중복 검사등을 들 수 있다.

유효성 검사를 위한 핸들러 함수

- 폼 페이지에서 이벤트가 발생했을때(<input onclick>을 클릭한 경우)의 유효성 검사를 위한 **매핑 메서드로 자바스크립트**를 이용하여 코드를 작성한다.

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

```
<form name="폼 이름">
    <input type="button" onclick="핸들러함수()">
</form>
```

유효성 검사(validation)

폼 페이지에 입력한 아이디와 비밀번호 출력하기

```
<title>유효성 검사</title>
</head>
<script type="text/javascript">
    function checkForm(){
        alert("아이디 : " + document.loginForm.userid.value + "\n"
            + "비밀번호 : " + document.loginForm.passwd.value)
    }
</script>
```

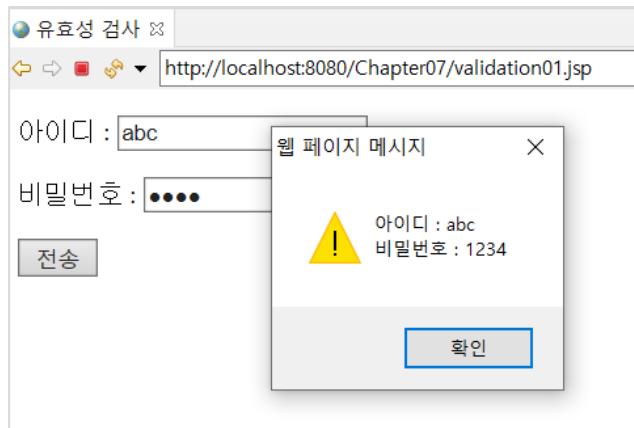
validation01.jsp

```
<form action="LoginProcess.jsp" method="post" name="LoginForm">
    <p>
        <label for="userid">아이디: </label>
        <input type="text" id="userid" name="userid">
    </p>
    <p>
        <label for="passwd">패스워드: </label>
        <input type="password" id="passwd" name="passwd">
    </p>
    <p><input type="submit" value="전송" onclick="checkForm()"></p>
</form>
```



유효성 검사(validation)

폼 페이지에 입력한 아이디와 비밀번호 출력하기

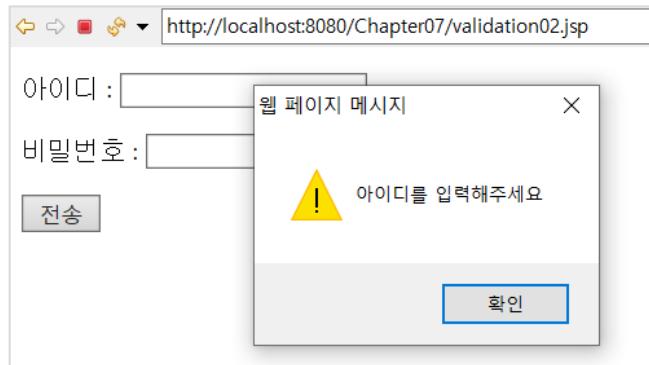


loginProcess.jsp

```
<title>로그인 처리</title>
</head>
<body>
    <p>로그인 되었습니다.</p>
</body>
</html>
```

유효성 검사(validation)

폼 페이지에 입력한 데이터 값의 유/무 검사하기 – [기본 유효성 검사]



validation02.jsp

```
<script type="text/javascript">
    function checkLogin(){
        let form = document.loginForm; // 폼 이름 변수 할당

        if(form.userid.value == ""){ // 아이디 입력 값이 없으면
            alert("아이디를 입력해주세요");
            form.userid.focus();
            return false;
        }else if(form.passwd.value == ""){
            alert("비밀번호를 입력해주세요");
            form.passwd.focus();
            return false;
        }else{
            form.submit(); // 폼을 전송함
        }
    }
</script>
```

※ return false 생략하면
action 페이지로 이동함

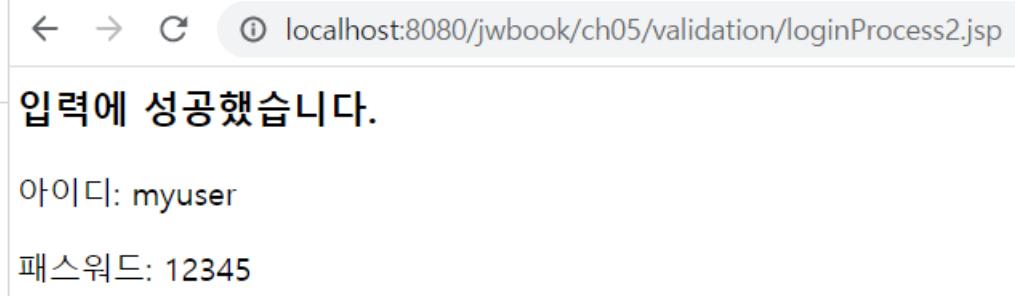
유효성 검사(validation)

loginProcess2.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

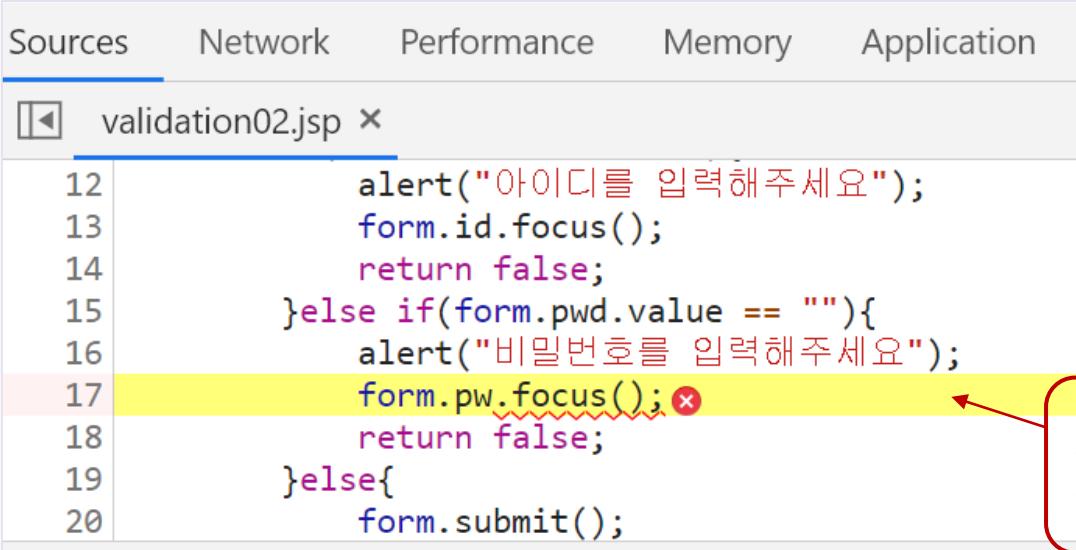
<%
    request.setCharacterEncoding("utf-8");

    String userid = request.getParameter("userid");
    String passwd = request.getParameter("passwd");
%>
<h3>입력에 성공했습니다.</h3>
<p>아이디: <%= userid %></p>
<p>패스워드: <%= passwd %></p>
```



유효성 검사(validation)

폼 페이지에 입력한 데이터 값의 유/무 검사하기



The screenshot shows the 'Sources' tab of a browser developer tools interface, specifically for a file named 'validation02.jsp'. The code is written in JavaScript and performs validation on form fields 'id' and 'pwd'. Lines 17 and 18 are highlighted with a yellow background, and line 18 contains a red squiggle under 'focus()' and a red circle with an 'X' at the end of the line, indicating a syntax error.

```
12     alert("아이디를 입력해주세요");
13     form.id.focus();
14     return false;
15 }else if(form.pwd.value == ""){
16     alert("비밀번호를 입력해주세요");
17     form.pw.focus(); ×
18     return false;
19 }else{
20     form.submit();
```

브라우저에서 오류 확인

자바스크립트 활용

회원가입 폼 만들기

회원가입

이름

이메일

localhost:8080 내용:
정말로 가입하시겠습니까?

이름

이메일

회원가입

가입 정보

이름: 안산

이메일: ansan@korea.kr

자바스크립트 활용

form/regForm.jsp

```
<h2>회원 가입</h2>
<hr>
<div id="regform">
    <form name="form1" action="" method="post">
        <label>이름</label><br>
        <input type="text" name="name" size="40">
        <hr>
        <label>이메일</label><br>
        <input type="text" name="email" size="40"><br>
        <p><button type="button" onclick="signUp()">가입</button></p>
    </form>
</div>

<div id="result">
    <h3>가입 정보</h3>
    <hr>
    <p>이름: <span id="rname"></span></p>
    <p>이메일: <span id="rmail"></span></p>
</div>
```



자바스크립트 활용

```
<style type="text/css">
    h2{
        background-color: green;
        text-align: center;
        padding: 15px 0;
        border-radius: 5px;
        color: #eee;
    }
    #regform{
        width: 50%;
        margin: 0 auto;
        background-color: lightgreen;
        padding: 15px 20px;
        border-radius: 10px;
    }
</style>
```

```
<script type="text/javascript">
    function signUp(){
        alert("정말로 가입하시겠습니까?");
        document.getElementById("regform").style.display = "none";
        document.getElementById("rname").innerHTML = document.form1.name.value;
        document.getElementById("rmail").innerHTML = document.form1.email.value;
        document.getElementById("result").setAttribute("style",
            "display:block; background: lightgreen; padding: 5px 20px;")
    }
</script>
```



유효성 검사(validation)

폼 페이지에 입력한 데이터 길이 확인하기

document.폼 이름.입력양식 이름.value.length

localhost:8080 내용:
아이디는 4-12자 이내로 입력 가능합니다.

validation.js

```
function checkLogin(){
    let form = document.loginForm;

    if(form.userid.value.length < 4 || form.userid.value.length > 12){
        alert("아이디는 4~12자 이내로 입력해주세요")
        form.userid.select();
        return false;
    }else if(form.passwd.value.length < 5){
        alert("비밀번호는 5자 이내로 입력해주세요")
        form.passwd.select();
        return false;
    }else{
        form.submit();
    }
}
```

유효성 검사(validation)

validation03.jsp

```
<title>유효성 검사</title>
<script src="../../resources/js/validation.js"></script>
</head>
<body>
    <form action="LoginProcess2.jsp" method="post" name="LoginForm">
        <p>
            <label for="userid">아이디: </label>
            <input type="text" id="userid" name="userid">
        </p>
        <p>
            <label for="passwd">패스워드: </label>
            <input type="password" id="passwd" name="passwd">
        </p>
        <p><input type="button" value="로그인" onclick="checkLogin()"></p>
    </form>
</body>
```



유효성 검사(validation)

폼(form)에 onsubmit 속성 사용하기

validation04.jsp

※ return 을 생략하면 action 페이지로 이동함

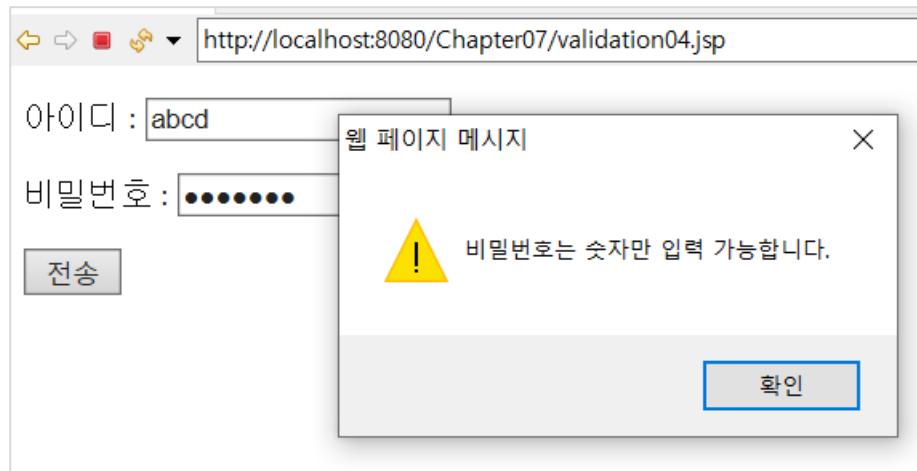
```
<script src="../../resources/js/validation.js"></script>
</head>
<body>
  <form action="LoginProcess2.jsp" method="post" name="LoginForm"
        onsubmit="return checkLogin()">
    <p>
      <label for="userid">아이디: </label>
      <input type="text" id="userid" name="userid">
    </p>
    <p>
      <label for="passwd">패스워드: </label>
      <input type="password" id="passwd" name="passwd">
    </p>
    <p><input type="submit" value="로그인"/></p>
  </form>
</body>
```



유효성 검사(validation)

폼 페이지에 입력한 데이터 값의 숫자 여부 확인하기

isNaN(document.폼 이름.입력양식 이름.value)



유효성 검사(validation)

자바스크립트 charAt() 메서드

The `charAt()` method returns the character at the specified index in a string.

The index of the first character is 0, the second character is 1, and so on.

Tip: The index of the last character in a string is `string.length-1`, the second last character is `string.length-2`, and so on (See "More Examples").

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to display the first character of the string  
"HELLO WORLD".</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
    var str = "HELLO WORLD";
    var res = str.charAt(0)
    document.getElementById("demo").innerHTML = res;
}
</script>

</body>
</html>
```

Click th
Try it
H

유효성 검사(validation)

```
function checkLogin(){
    let form = document.loginForm;
    let id = form.userid.value;
    let pw = form.passwd.value;

    if(id==""){
        alert("아이디를 입력해 주세요");
        form.id.focus();
        return false;
    }
    for(var i=0; i<id.length; i++){
        var ch = id.charAt(i);

        if((ch<'a' || ch>'z') && (ch>'A' || ch<'Z') && (ch>'0' || ch<'9')){
            alert("아이디는 영문 소문자만 가능합니다.");
            form.userid.select();
            return false;
        }
    }
}
```

checklogin.js



유효성 검사(validation)

checklogin.js

```
if(pw==""){
    alert("비밀번호를 입력해주세요");
    form.passwd.focus();
    return false;
}
if(isNaN(pw)){
    alert("비밀번호는 숫자만 입력해주세요");
    form.passwd.select();
    return false;
}

form.submit();
}
```



정규 표현식(Regular Expression)

정규 표현식(Regular Expression)

- 특정한 규칙을 가진 문자열의 집합을 표현하는데 사용하는 형식 언어이다.
즉, 문자열의 특정 형태를 찾아내기 위해 패턴으로 표현한 수식이다.
전화번호, 이메일과 같이 데이터 형식의 패턴이 일정한 데이터를 검사하는 데 이용한다.
- 사용 형식

let 변수 이름 = /정규 표현식/[Flag]

let 변수 이름 = new RegExp(정규 표현식, [Flag])

Flag	설 명
i	Ignore Case: 문자열의 대문자와 소문자를 구별하지 않고 검출함
g	Global: 문자열 내의 모든 패턴을 검출함
gi	모든 패터를 검출하면 대소문자 구별하지 않고 검출함
m	Multi line: 문자열에 줄바꿈이 있는지 검출한다.

정규 표현식(Regular Expression)

- 기본 메타 문자 및 문자 클래스

메타문자	설 명
x	문자열이 x로 시작함
$^$$	문자열이 x로 종료됨
$x?$	x가 존재하거나 존재하지 않음
x^*	x가 0번 이상 반복됨
[xy]	x 또는 y를 찾음
\x	x 문자를 찾음
\d	숫자(decimal)를 찾음
\s	공백(space)을 찾음
\w	알파벳+숫자(word)를 찾음
{3, 4}	3번에서 4번 반복함

정규 표현식(Regular Expression)

- 자주 사용하는 패턴

표현식	설명
/^0-9]*\$/	숫자만
/^a-zA-Z]*\$/	영문자만
/^가-힣]*\$/	한글만
/^a-zA-Z0-9]*\$/	영문자와 숫자만
\d{3}-\d{3,4}-\d{4}/	휴대폰
/[0-9a-zA-Z]@[0-9a-zA-Z].[a-zA-Z]{2,3}/	이메일
\d{6}[-][1-4]{6}	주민등록번호
\d{3}[-]\d{2}	우편번호

정규 표현식(Regular Expression)

- 자주 사용하는 메소드

메소드	설명
<code>test()</code>	매개변수 값으로 전달되는 문자열이 정규 표현식에 부합한지 판단하여 true/false를 반환함
<code>exec()</code>	매개변수 값으로 전달되는 문자열이 정규 표현식에 부합된 문자열을 추출하여 반환함
<code>search()</code>	정확히 일치하는 문자열의 위치를 반환함
<code>match()</code>	정확히 일치하는 문자열을 반환함

정규 표현식(Regular Expression)

정규 표현식(Regular Expression) - JavaScript

https://www.w3schools.com/js/js_regexp.asp

Regular Expression Modifiers

Modifiers can be used to perform case-insensitive more global searches:

Modifier	Description	Try it
i	Perform case-insensitive matching	Try it »
g	Perform a global match (find all matches rather than stopping after the first match)	Try it »
m	Perform multiline matching	Try it »

Regular Expression Patterns

Brackets are used to find a range of characters:

Expression	Description	Try it
[abc]	Find any of the characters between the brackets	Try it »
[0-9]	Find any of the digits between the brackets	Try it »
(x y)	Find any of the alternatives separated with	Try it »



match() 함수

match()는 정확히 일치하는 문자열을 반환

```
<p>Do a search for "ain":</p>

<p id="demo"></p>

<script>
let text = "The rain in SPAIN stays mainly in the plain";
let result = text.match("ain");

document.getElementById("demo").innerHTML = result;
</script>
```

```
<p>match() searches for a match against a regular expression.</p>

<p>Do a global, case-insensitive, search for "ain":</p>

<p id="demo"></p>

<script>
let text = "The rain in SPAIN stays mainly in the plain";
let result = text.match(/ain/gi);

document.getElementById("demo").innerHTML = result;
</script>
```



search() 함수

```
<h1>JavaScript Strings</h1>
<h2>The search() Method</h2>
```

```
<p>search() searches a string for a value and returns the position
</p>
```

```
<p id="demo"></p>
```

```
<script>
let text = "Mr. Blue has a blue house"
let position = text.search(/blue/i);
```

```
document.getElementById("demo").innerHTML = position;
</script>
```

search()는 정확히 일치하는 문자열의 위치를 반환

JavaScript Strings

The search() Method

search() searches a string for a value and returns the position of first the match:

정규 표현식(Regular Expression)

JavaScript test() Method

The test() method tests for a match in a string.

→ **test()**는 문자열이 일치하는지 여부를 true/false로 반환.

This method returns true if it finds a match, otherwise it returns false.

```
<!DOCTYPE html>
<html>
<body>

<p>The test() method returns true if it finds a match, otherwise it
returns false.</p>
<p>Click the button to search a string for the character "e".</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
    var str = "The best things in life are free";
    var patt = new RegExp("e");
    var res = patt.test(str);
    document.getElementById("demo").innerHTML = res;
}
</script>

</body>
</html>
```

The test
Click the
Try it
true



유효성 검사(validation)

정규 표현식 사용 예제

localhost:8080/jwbook/ch05/reg_exp/regexp1.jsp

이름: 1김기용 전송

localhost:8080 내용:
이름은 숫자로 시작할 수 없습니다.

확인

유효성 검사(validation)

```
<title>Validation</title>
<script>
    function checkForm(){
        //이름은 숫자로 시작할 수 없음
        let str = document.frm.name.value;      //입력 양식 값 변수 할당
        let regExp = /^[a-zA-Z가-힣]/; //정규표현식 변수 선언(시작 - ^사용)

        if(!regExp.test(str)){
            alert("이름은 숫자로 시작할 수 없습니다.");
            return false;
        }
    }
</script>
</head>
<body>
    <form action="" method="post" name="frm">
        <p>이름: <input type="text" name="name">
        <input type="submit" value="전송" onclick="checkForm()">
    </form>
</body>
</html>
```

regexp.jsp



Session 연습문제

The screenshot shows a web browser window with the URL `localhost:8080/jwbook/ch05/reg_exp/validation_test.jsp`. The page title is "회원가입". On the left, there is a form with three input fields: "아이디" containing "cloud", "비밀번호" containing ".....", and "이름" (name) empty. A "가입하기" (Join) button is at the bottom. A tooltip message "localhost:8080 내용: 비밀번호는 아이디를 포함할 수 없습니다." (localhost:8080 Content: Password cannot contain the ID) appears over the password field. On the right, a blue arrow points to a box containing the results: "아이디 : cloud", "비밀번호 : clou1234", and "이 름 : 안산".

1. validation_test.jsp 파일 생성

- 1) 아이디, 비밀번호가 입력되지 않으면 "아이디(또는 비밀번호)가 입력해 주세요'라는 메시지 창이 뜨도록 작성한다.
- 2) 비밀번호에 아이디 값을 포함하면 "비밀번호는 아이디를 포함할 수 없습니다."라는 메시지 창이 뜨도록 작성한다.
- 3) 이름이 입력되지 않으면 "이름을 입력해 주세요'라는 메시지 창이 뜨도록 작성한다.

2. test_process.jsp 파일 생성하여 요청 파라미터값을 출력한다.

Session 연습문제

validation_test.jsp

```
<h3>회원 가입</h3>
<form action="test_process.jsp" method="post" name="Member">
    <p>아이디: <input type="text" name="id">
    <p>비밀번호: <input type="password" name="passwd">
    <p>이 름: <input type="text" name="name">
    <p> <button type="button" onclick="checkMember()">가입하기</button>
</form>
```



유효성 검사(validation)

회원가입 폼에 입력한 데이터 형식 유효성 검사 – 정규 표현식

The image displays two side-by-side screenshots of a web browser window, both showing the URL `localhost:8080/jwbook/ch05/reg_exp/joinForm.jsp`. The left screenshot shows validation for a phone number input, and the right screenshot shows validation for an email address input.

Left Screenshot (Phone Number Validation):

- Form Fields:**
 - 아이디: `cloud100`
 - 비밀번호: `.....`
 - 이름: `김기용`
 - 연락처: `010` (dropdown) - `10` (input) - `5678` (input)
 - 이메일: `.....`
 - 가입하기
- Validation Message:** `localhost:8080 내용:
연락처 입력을 확인해주세요`
- Buttons:** 확인 (Confirm)

Right Screenshot (Email Address Validation):

- Form Fields:**
 - 아이디: `member1`
 - 비밀번호: `.....`
 - 이름: `신유빈`
 - 연락처: `010` (dropdown) - `1234` (input) - `5678` (input)
 - 이메일: `ping@pong`
 - 가입하기
- Validation Message:** `localhost:8080 내용:
이메일 입력을 확인해주세요`
- Buttons:** 확인 (Confirm)

유효성 검사(validation)

회원 가입 폼에 입력한 데이터 형식 유효성 검사 – 정규 표현식

regexp_join.jsp

```
<h3>회원 가입</h3>
<form action="validation_process.jsp" method="post" name="Member">
    <p>아이디: <input type="text" name="id">
    <p>비밀번호: <input type="password" name="passwd">
    <p>이 름: <input type="text" name="name">
    <p>연락처: <select name="phone1">
        <option value="010">010</option>
        <option value="02">02</option>
        <option value="031">031</option>
        <option value="041">041</option>
    </select> - <input type="text" maxlength="4" size="4" name="phone2">
        - <input type="text" maxlength="4" size="4" name="phone3">
    <p> 이메일: <input type="text" name="email">
    <p> <button type="button" onclick="checkMember()">가입하기</button>
</form>
```



유효성 검사(validation)

```
<script>
    function checkMember(){
        //정규 표현식 변수 할당
        let regExpId = /^[a-zA-Zㄱ-ㅎ가-힣]/;
        let regExpPasswd = /^[0-9]*$/;
        let regExpName = /^[가-힣]*$/;
        let regExpPhone = /\d{3}-\d{3,4}-\d{4}/;
        //let regExpEmail = /^[0-9a-zA-Z]*@[0-9a-zA-Z]*\.[a-zA-Z]{2,3}$/;
        let regExpEmail
            = /^[0-9a-zA-Z]([-_]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_]?[0-9a-zA-Z])*\. [a-zA-Z]{2,3}$/;
        //특수문자 포함 - x? (x가 있거나 없거나), \.(점을 찾음)

        //폼 변수 할당
        let form = document.Member;

        //입력 양식 값 변수 할당
        let id = form.id.value;
        let pw = form.passwd.value;
        let name = form.name.value;
        let phone = form.phone1.value + "-" + form.phone2.value +
                    "-" + form.phone3.value;
        let email = form.email.value;
```

regexp_join.jsp



유효성 검사(validation)

```
if(id == "" || !regExpId.test(id)){
    alert("아이디는 문자로 시작해주세요!");
    form.id.select();
    return false;
}
if(pw == "" || !regExpPasswd.test(pw)){
    alert("비밀번호는 숫자만 입력해주세요!");
    return false;
}
if(name == "" || !regExpName.test(name)){
    alert("이름은 한글만 입력해주세요!");
    return false;
}
if(!regExpPhone.test(phone)){
    alert("연락처 입력을 확인해주세요");
    return false;
}
if(!regExpEmail.test(email)){
    alert("이메일 입력을 확인해주세요");
    return false;
}
form.submit(); //폼을 서버 페이지로 전송
}
```

regexp_join.jsp



유효성 검사(validation)

validation01_process.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

<% request.setCharacterEncoding("utf-8"); %>

<p>아이디 : <%=request.getParameter("id") %>
<p>비밀번호 : <%=request.getParameter("passwd") %>
<p>이 름 : <%=request.getParameter("name") %>
<p>연락처 : <%=request.getParameter("phone1") %>
    - <%=request.getParameter("phone2") %>
    - <%=request.getParameter("phone3") %>
<p>이메일 : <%=request.getParameter("email") %>
```



유효성 검사(validation)

비밀번호 숫자와 영문자의 조합으로 입력하기

The screenshot shows a web browser window with the URL `localhost:8080/jwbook/ch05/reg_exp/joinForm2.jsp`. The page title is "회원가입". The form fields are:

- 아이디:
- 비밀번호:
- 이 름:
- 가입하기
- 확인

A tooltip-like message box is displayed over the password field, containing the text:

localhost:8080 내용:
비밀번호는 영문자, 숫자, 특수문자 포함 8~12자로 입력해주세요

유효성 검사(validation)

```
function checkMember(){
    //정규 표현식 변수 할당
    let regExpId = /^[a-zA-Zㄱ-ㅎ가-힣]/;
    let regExpPasswd1 = /[A-Za-z0-9]/;
    let regExpPasswd2 = /[^!@#$%^&*+-/_]/;

    let form = document.Member;    //폼 변수
    //입력 양식 값 변수 할당
    let id = form.id.value;
    let pw = form.passwd.value;
    let name = form.name.value;

    if(id == "" || !regExpId.test(id)){
        alert("아이디는 문자로 시작해주세요!");
        form.id.select();
        return false;
    }
    if(pw.length < 8 || pw.length > 12 || !regExpPasswd1.test(pw)
       || !regExpPasswd2.test(pw)){
        alert("비밀번호는 영문자, 숫자, 특수문자 포함 8~12자로 입력해주세요");
        return false;
    }
    if(name == ""){
        alert("이름을 입력해주세요!");
        return false;
    }
    form.submit();    //폼을 서버 페이지로 전송
}
```

