제13장
유효성 검사

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 form 페이지에서 이벤트가 발생했을 때(<submit>를 클릭한 경우)의 유효성 검사를 위해 매핑하는 메서드
  - **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

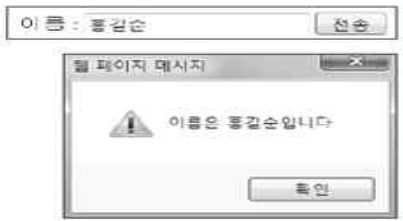
```
<script type="text/javascript">
  function 핸들러 함수(){
     var str = document.폼 이름.입력항목 이름.value;
  }
</script>

<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>
```



2. 기본 유효성 검사

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.

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

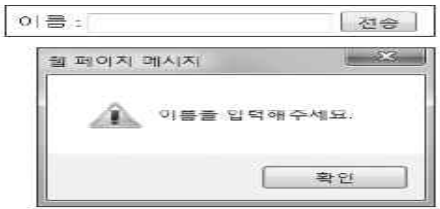
```
document.폼 이름.입력양식 이름.value==""
```

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">
    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>
```



2. 기본 유효성 검사

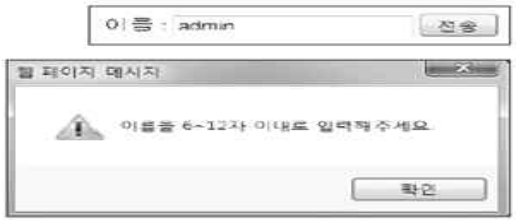
- **데이터 길이 확인하기**
  - 회원 가입 페이지에서 아이디, 비밀번호 등과 같은 입력 데이터의 제한 길이를 검사하는 것
    - 예를 들면 입력 데이터의 조건으로 아이디와 비밀번호는 4~12자 이내로 영어와 숫자를 혼합해서 입력할 것, 첫 문자는 숫자로 시작할 수 없음 등을 검사하는 것

```
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>
```



2. 기본 유효성 검사

숫자 여부 확인하기

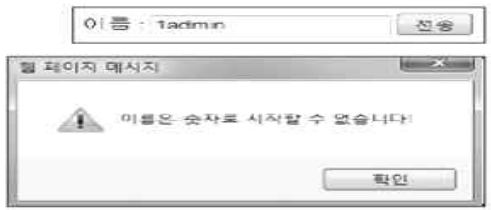
- 숫자 여부는 isNaN() 함수를 활용하여 검사
- isNaN
  - isNotaNumber의 약자
  - isNaN() 함수의 인자 값이 숫자이면 false를 반환하고 숫자가 아니면 true를 반환

```
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>
```



1. 전체 폼 초기화
document.form.reset()
2. 선택 초기화
document.form.name.select();
document.selection.clear();

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>
```

3. 데이터 형식 유효성 검사

정규 표현식의 표현 방법

- 기본 메타 문자의 종류

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

3. 데이터 형식 유효성 검사

정규 표현식의 표현 방법

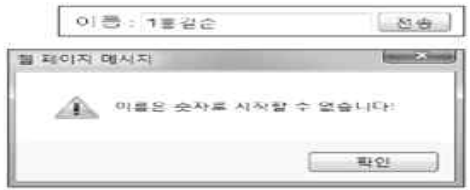
- 문자 클래스의 종류

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.
₩^	^(특수문자)을 식에 문자 자체로 포함합니다.
₩b	문자와 공백 사이의 문자를 찾습니다.
₩B	공백을 제외한 문자와 문자 사이의 문자를 찾습니다.
₩d	숫자를 찾습니다.
₩D	숫자가 아닌 값을 찾습니다.
₩s	공백 문자를 찾습니다.
₩S	공백이 아닌 문자를 찾습니다.

문자 클래스	설명
₩t	Tab 문자를 찾습니다.
₩v	Vertical Tab 문자를 찾습니다.
₩w	알파벳 + 숫자 + _을 찾습니다.
₩W	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-z|A-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>
```



감사합니다.