

JSP 웹 프로그래밍

6 폼 태그

쉽게 배우는 JSP 웹 프로그래밍

1

폼 처리의 개요

2

form 태그의 기능과 사용법

3

input 태그의 기능과 사용법

4

select 태그의 기능과 사용법

5

textarea 태그의 기능과 사용법

6

폼 데이터 처리하기

7

[웹 쇼핑몰] 상품 등록 페이지 만들기

학습목표

- 폼의 개념과 특징을 이해합니다.
- 폼 태그 구성 요소의 사용법을 익힙니다.
- 폼 데이터를 처리하는 방법을 이해합니다.
- 폼 태그를 이용하여 웹 쇼핑몰의 상품 등록 페이지를 출력합니다.

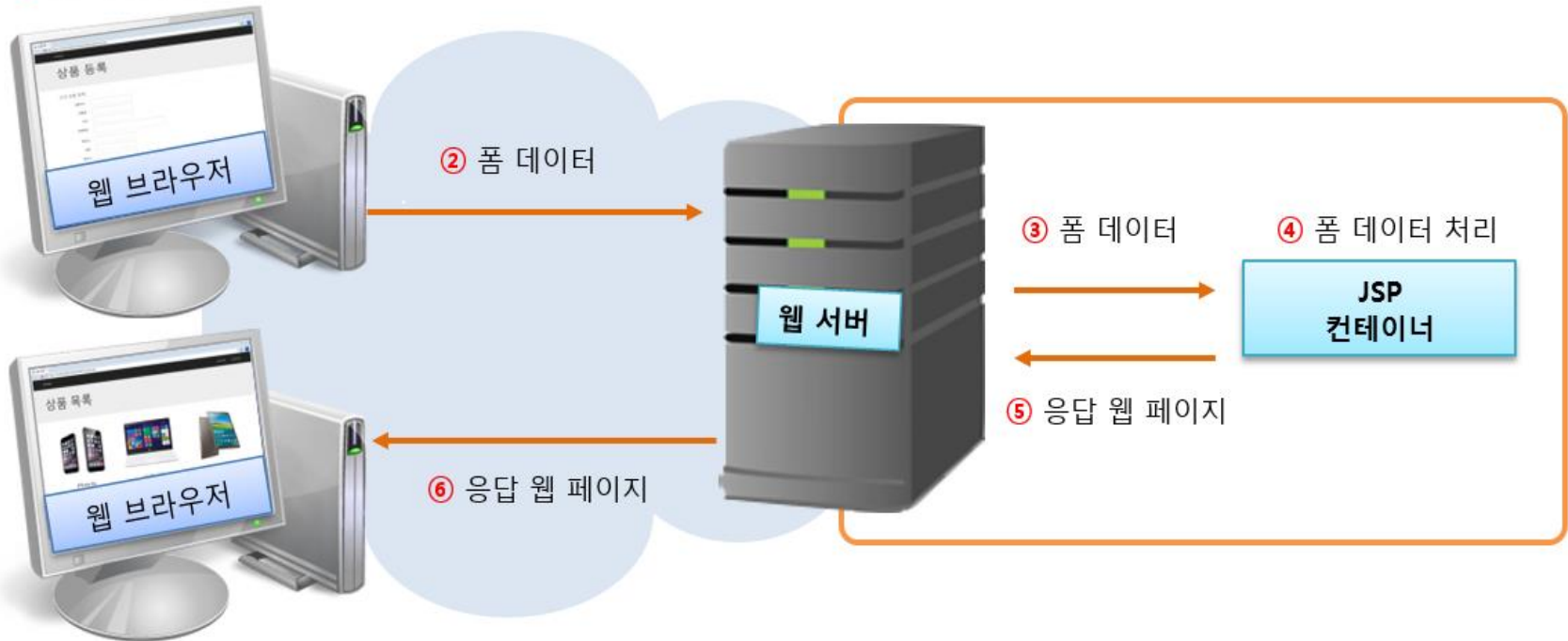
❖ 폼(form)

- 사용자가 웹 브라우저를 통해 입력된 모든 데이터를 한 번에 웹 서버로 전송하는 양식
 - 전송한 데이터는 웹 서버가 처리하고 처리 결과에 따라 다른 웹 페이지를 보여줌
- 사용자와 웹 애플리케이션이 상호 작용하는 중요한 기술 중 하나
- 사용자가 어떤 내용을 원하는지, 사용자의 요구 사항이 무엇인지 파악할 때 가장 많이 사용하는 웹 애플리케이션의 필수적인 요소

1. 폼 처리의 개요

❖ 폼 데이터 처리 과정

① 폼 데이터 입력



⑦ 웹 페이지 출력

1. 폼 처리의 개요

❖ 폼을 구성하는 태그의 종류

태그	설명	태그	설명
form	폼을 정의하는 태그로 최상위 태그입니다.	select	항목을 선택할 수 있는 태그입니다.
input	사용자가 입력할 수 있는 태그입니다.	textarea	여러 줄을 입력할 수 있는 태그입니다.

2. form 태그의 기능과 사용법

❖ form 태그

- 사용자가 다양한 정보를 입력하고 서로 전달할 때 사용하는 태그
- 단독으로 쓰이지 않고 사용자가 다양한 정보를 입력하는 양식을 포함하는 최상위 태그

```
<form 속성1="값1" [속성2="값2" ...]>
    //다양한 입력 양식 태그(<input>, <select>, <textarea>)
</form>
```

- 속성을 이용하여 폼 데이터를 전송할 때 어디로 보낼지, 어떤 방식으로 보낼지 설정
- form 태그의 모든 속성은 필수가 아니라 선택적으로 사용

속성	설명
action	폼 데이터를 받아 처리하는 웹 페이지의 URL을 설정합니다.
method	폼 데이터가 전송되는 HTTP 방식을 설정합니다.
name	폼을 식별하기 위한 이름을 설정합니다.
target	폼 처리 결과의 응답을 실행할 프레임을 설정합니다.
enctype	폼을 전송하는 콘텐츠 MIME 유형을 설정합니다.
accept-charset	폼 전송에 사용할 문자 인코딩을 설정합니다.

2. form 태그의 기능과 사용법(속성 : method)

❖ GET 방식과 POST 방식의 차이

구분	GET방식	POST방식
전송 형태	이름과 값의 형태로 URL에 포함되어 전송	HTTP 헤더 속에 감춰서 전송
전송량	제한적(4,960B)	제한없음
속도	빠름	느림
보안	없음	있음

■ POST 방식

- 개인정보 등의 전송에 사용
- 지정된 리소스에서 데이터를 처리할 때(쓰고, 수정하고, 삭제할 때)

■ GET 방식

- 지정된 리소스에서 데이터를 요청할 때(즉 읽을 때)

2. form 태그의 기능과 사용법

[form 태그 사용 예]

```
<html>
<head>
<title>Form Processing</title>

</head>
<body>
    <form action="http://localhost:8080/ch06/process.jsp" method="get"
        name="login", accept-charset="utf-8">
        ...(생략)...
    </form>
</body>
</html>
```


3. input 태그의 기능과 사용법

❖ input 태그

- 사용자가 텍스트 입력이나 선택 등을 다양하게 할 수 있도록 공간을 만드는 태그
- 종료 태그 없이 단독으로 사용할 수 있음

```
<input 속성1="값1" [속성2="값2" ...]>
```

- input 태그의 기본 속성

속성	속성 값	설명
type	text	기본 값으로 한 줄의 텍스트를 입력할 때 사용합니다.
	radio	라디오 버튼으로 열거된 것 중 하나만 선택할 때 사용합니다.
	checkbox	체크 박스로 열거된 것 중 다중 선택을 할 때 사용합니다.
	password	암호를 입력할 때 사용합니다.
	hidden	보이지 않게 숨겨서 값을 전송할 때 사용합니다.
	file	파일 업로드를 위한 파일을 선택할 때 사용합니다.
	button	버튼 모양을 출력할 때 사용합니다.
	reset	폼에 입력된 값을 모두 초기화할 때 사용합니다.
	submit	폼에 입력된 값을 모두 서버에 전송할 때 사용합니다.
name	텍스트	입력 양식을 식별하는 이름을 설정합니다.
value	텍스트	입력 양식의 초기값을 설정합니다.

3. input 태그의 기능과 사용법

[input 태그 사용 예]

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

<head>
<title>Form Processing</title>
</head>
<body>
  <form action="#" method="get" >
    <p>아 이 디: <input type="text" name="id">
    <p>비밀번호: <input type="password" name="passwd">
    <p><input type="submit" value="전송">
  </form>
</body>
</html>
```



아 이 디:

비밀번호:

3. input 태그의 기능과 사용법

예제 6-1 form 태그와 input 태그로 간단한 회원 가입 양식 만들기

JSPBook/WebContent/ch06/form01.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Form Processing</title>
05 </head>
06 <body>
07     <h3>회원 가입</h3>
08     <form action="#" name="member" method="post">
09         <p> 아이디 : <input type="text" name="id"> <input type="button"
10             value="아이디 중복 검사">
11         <p> 비밀번호 : <input type="password" name="passwd">
12         <p> 이름 : <input type="text" name="name">
13         <p> 연락처 : <input type="text" maxlength="4" size="4" name="phone1"> -
14             <input type="text" maxlength="4" size="4" name="phone2"> -
15             <input type="text" maxlength="4" size="4" name="phone3">
16         <p> 성별 : <input type="radio" name="sex" value="남성" checked>남성
17             <input type="radio" name="sex" value="여성" >여성
18         <p> 취미 : 독서<input type="checkbox" name="hobby1" checked>
19             운동<input type="checkbox" name="hobby2">
20             영화<input type="checkbox" name="hobby3">
21         <p> <input type="submit" value="가입하기">
22             <input type="reset" value="다시 쓰기">
23     </form>
24 </body>
25 </html>
```

Form Processing

http://localhost:8080/JSPBook/ch06/form01.jsp

회원 가입

아이디 :

비밀번호 :

이름 :

연락처 : - -

성별 : ☒ 남성 ☐ 여성

취미 : 독서 ☒ 운동 ☐ 영화 ☐

4. select 태그의 기능과 사용법

❖ select 태그

- 여러 개의 항목이 나타나는 목록 상자에서 항목을 선택하는 태그
- 시작 태그와 종료 태그가 있으며, 리스트 박스에 여러 항목을 추가 삽입하기 위해 반드시 option 태그를 포함해야 함.

```
<select 속성1="값1" [속성2="값2" ...]>
  <option 속성1="값" [속성1]> 항목1</option>
  <option 속성2="값" [속성2]> 항목2</option>
  ...(생략)...
</select>
```

■ select 태그의 속성

속성	속성 값	설명
name	텍스트	목록 상자의 이름을 설정합니다.
size	숫자	한 번에 표시할 항목의 개수를 설정합니다.
multiple		다중 선택이 가능하도록 합니다. [Ctrl] 키를 눌러 목록 상자의 항목을 다중 선택합니다.

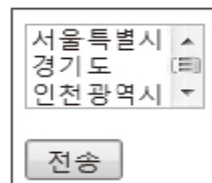
option 태그의 속성

속성	속성 값	설명
value	텍스트	항목의 값을 설정합니다.
selected		해당 항목을 초깃값으로 선택합니다.
disabled		항목을 비활성화합니다.

4. select 태그의 기능과 사용법

[select 태그 사용 예]

```
<? page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Form Processing</title>
</head>
<body>
  <form action="#" method="get">
    <p> <select name="city" size="3">
      <option value="서울시">서울특별시</option>
      <option value="경기도">경기도</option>
      <option value="인천시">인천광역시</option>
      <option value="충청도">충청도</option>
      <option value="전라도">전라도</option>
      <option value="경상도">경상도</option>
    </select>
    <p> <input type="submit" value="전송">
  </form>
</body>
</html>
```



4. select 태그의 기능과 사용법

예제 6-2 select 태그로 [예제 6-1] 회원 가입 양식의 연락처 수정하기

JSPBook/WebContent/ch06/form02.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02     ... (생략) ...
03     <p> 연락처 : <select name="phone1">
04         <option value="010">010</option>
05         <option value="011">011</option>
06         <option value="016">016</option>
07         <option value="017">017</option>
08         <option value="019">019</option>
09     </select> - <input type="text" maxlength="4" size="4"
10     name="phone2"> - <input type="text" maxlength="4" size="4"
11     name="phone3">
12     ... (생략) ...
```

Form Processing

http://localhost:8080/JSPBook/ch06/form02.jsp

회원 가입

아이디 : 아이디 중복 검사

비밀번호 :

이름 :

연락처 : 010 - -

성별 : ☒ 남성 ☐ 여성

취미 : 독서 ☒ 운동 ☐ 영화 ☐

가입하기 다시 쓰기

5. textarea 태그의 기능과 사용법

❖ textarea 태그

- 여러 줄의 텍스트를 입력할 수 있는 태그
- 기본 값은 <textarea>와 </textarea> 태그 사이에 설정
- 입력 폼 안에 사용된 태그와 띄어쓰기가 그대로 출력됨

```
<textarea cols="너비 값" rows="높이 값">  
...(생략)...  
</textarea>
```

■ textarea 태그의 속성

속성	속성 값	설명
name	텍스트	이름을 설정합니다.
cols	숫자	입력할 텍스트 영역의 너비(열 크기)를 설정합니다.
rows	숫자	입력할 텍스트 영역의 높이(행 크기)를 설정합니다.
wrap	off	줄 바꿈을 설정합니다. wrap="off": 줄 바꿈을 하지 않고 문장을 입력할 때 수평 스크롤바가 생기고 옆으로 계속 문장이 입력됩니다.
	soft	wrap="soft": [Enter] 를 누르지 않아도 텍스트라인 끝에서 자동으로 행이 바뀝니다.
	hard	wrap="hard": soft 상태와 비슷하며 실제 내용을 서버에 전송할 때 캐리지 리턴 문자를 전달합니다.

5. textarea 태그의 기능과 사용법

[textarea 태그 사용 예]

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

```
<head>
<title>Form Processing</title>
</head>
```

```
<body>
  <form action="#" method="get">
    <textarea name="comment" cols="30" rows="3"></textarea>
    <p> <input type="submit" value="전송" />
      <input type="reset" value="취소" />
    </form>
  </body>
</html>
```


5. textarea 태그의 기능과 사용법

예제 6-3 textarea 태그로 [예제 6-2]의 회원 가입 양식에 가입 인사 추가하기

JSPBook/WebContent/ch06/form03.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02     ... (생략) ...
03     <p> 취미 : 독서<input type="checkbox"
04         name="hobby1" checked>
05         운동<input type="checkbox"
06         name="hobby2">
07         영화<input type="checkbox"
08         name="hobby3">
09     <p> <textarea name="comment" cols="30"
10         rows="3" placeholder="가입 인사를 입력해주세요">/textarea>
11     ... (생략) ...
```

Form Processing

http://localhost:8080/JSPBook/ch06/form03.jsp

회원 가입

아이디 : 아이디 중복 검사

비밀번호 :

이름 :

연락처 : 010 - -

성별 : ☒ 남성 ☐ 여성

취미 : 독서 ☒ 운동 ☐ 영화 ☐

가입 인사를 입력해주세요

가입하기 다시 쓰기

6. 폼 데이터 처리하기

❖ 요청 파라미터의 값 받기

- request 내장 객체는 웹 브라우저가 서버로 보낸 요청에 대한 다양한 정보를 담고 있어 `getParameter()` 메소드를 이용하여 요청 파라미터의 값을 얻을 수 있음

```
String 변수 = request.getParameter(요청 파라미터 이름);
```

6. 폼 데이터 처리하기

[체크 박스의 데이터를 전달받는 예]

```
//index.jsp
<%@ page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Form Processing</title>

</head>
<body>
    <form action="process.jsp" method="POST">
        <p> 독서<input type="checkbox" name="reading">
            운동<input type="checkbox" name="exercise">
            영화<input type="checkbox" name="movie">
        <p> <input type="submit" value="전송">
    </form>
</body>
</html>
```

독서	<input checked="" type="checkbox"/>
운동	<input type="checkbox"/>
영화	<input checked="" type="checkbox"/>

전송

```
//process.jsp
<%@ page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Form Processing</title>
</head>
<body>
    <p> 독서 : <%=request.getParameter("reading")%>
    <p> 운동 : <%=request.getParameter("exercise")%>
    <p> 영화 : <%=request.getParameter("movie")%>
</body>
</html>
```

독서	: on
운동	: null
영화	: on

6. 폼 데이터 처리하기

예제 6-4 [예제 6-3]의 회원 가입 양식에서 폼 데이터 전송받기

JSPBook/WebContent/ch06/form04.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 ... (생략) ...
03     <form action="form04_process.jsp" name="member" method="post">
04         ... (생략) ...
```

6. 폼 데이터 처리하기

JSPBook/WebContent/ch06/form04_process.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Form Processing</title>
05 </head>
06 <body>
07     <%
08         request.setCharacterEncoding("UTF-8");
09
10         String id = request.getParameter("id");
11         String passwd = request.getParameter("passwd");
12         String name = request.getParameter("name");
13         String phone1 = request.getParameter("phone1");
14         String phone2 = request.getParameter("phone2");
15         String phone3 = request.getParameter("phone3");
16         String sex = request.getParameter("sex");
17         String hobby1 = request.getParameter("hobby1");
18         String hobby2 = request.getParameter("hobby2");
19         String hobby3 = request.getParameter("hobby3");
20         String comment = request.getParameter("comment");
21     %>
22     <p> 아이디 : <%=id%>
23     <p> 비밀번호 : <%=passwd%>
24     <p> 이름 : <%=name%>
25     <p> 연락처 : <%=phone1%><%=phone2%><%=phone3%>
26     <p> 성별 : <%=sex%>
27     <p> 취미 : <%=hobby1%> <%=hobby2%> <%=hobby3%>
28     <p> 가입 인사 : <%=comment%>
29 </body>
30 </html>
```

6. 폼 데이터 처리하기

예제 6-5 [예제 6-4]의 회원 가입 양식에서 폼 데이터 전송받기

JSPBook/WebContent/ch06/form05.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 ... (생략) ...
03     <form action="form05_process.jsp" name="member" method="post">
04 ... (생략) ...
05         <p> 취미 : 독서<input type="checkbox" name="hobby" value="독서" checked>
06             운동<input type="checkbox" name="hobby" value="운동" >
07             영화<input type="checkbox" name="hobby" value="영화">
08         <p> <textarea name="comment" cols="30" rows="3" placeholder="가입 인
           사를 입력해주세요">X</textarea>
09 ... (생략) ...
```

6. 폼 데이터 처리하기

JSPBook/WebContent/ch06/form05_process.jsp

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>Form Processing</title>
05 </head>
06 <body>
07     <%
08         request.setCharacterEncoding("UTF-8");
09
10         String id = request.getParameter("id");
11         String passwd = request.getParameter("passwd");
12         String name = request.getParameter("name");
13         String phone = request.getParameter("phone");
14         String phone2 = request.getParameter("phone2");
15         String phone3 = request.getParameter("phone3");
16         String sex = request.getParameter("sex");
17         String[] hobby = request.getParameterValues("hobby");
18         String comment = request.getParameter("comment");
19     %>
```

Form Processing

/JSPBook/ch06/form05_process.jsp

아이디 : admin

비밀번호 : 1234

이름 : 홍길순

연락처 : 010-9002-7370

성별 : 여성

취미 : 독서 영화

가입 인사 : 안녕하세요. 홍길순 가입합니다.

6. 폼 데이터 처리하기

```
20     <p> 아이디 : <%=id%>
21     <p> 비밀번호 : <%=passwd%>
22     <p> 이름 : <%=name%>
23     <p> 연락처 : <%=phone%><%=phone2%><%=phone3%>
24     <p> 성별 : <%=sex%>
25     %>
26     <p> 취미 : <%
27         if (hobby != null) {
28             for (int i = 0; i < hobby.length; i++) {
29                 out.println(" " + hobby[i]);
30             }
31         }
32     %>
33     <p> 가입 인사 : <%=comment%>
34 </body>
35 </html>
```


6. 폼 데이터 처리하기

❖ 요청 파라미터의 전체 값 받기

- 요청 파라미터를 설정하지 않아도 모든 값을 전달받을 수 있습니다. 또한 텍스트 박스, 라디오 버튼, 드롭다운 박스와 같은 다양한 유형에 대해 한 번에 폼 데이터를 전달받을 수 있음.
- 폼 데이터의 일괄 처리 메소드

메소드	형식	설명
getParameterNames()	java.util.Enumeration	모든 입력 양식의 요청 파라미터 이름을 순서에 상관없이 Enumeration(열거형) 형태로 전달받습니다.
hasMoreElements()		Enumeration(열거형) 요소가 있으면 true를 반환하고, 그렇지 않으면 false를 반환합니다.
nextElement()		Enumeration(열거형) 요소를 반환합니다.

6. 폼 데이터 처리하기

[체크 박스의 전체 데이터를 전달받는 예]

```
//process.jsp
<%@ page contentType="text/html; charset=utf-8"%>
<%@ page import="java.io.*,java.util.*"%>
<html>
<head>
<title>Form Processing</title>
</head>
<body>
    <table border="1">
        <tr>
            <th>요청 파라미터 이름</th>
            <th>요청 파라미터 값</th>
        </tr>
        <%
            Enumeration paramNames = request.getParameterNames();
            while (paramNames.hasMoreElements()) {
                String name = (String) paramNames.nextElement();
                out.print("<tr><td>" + name + " </td>\n");
                String paramValue = request.getParameter(name);
                out.println("<td> " + paramValue + "</td></tr>\n");
            }
        %>
    </table>
</body>
</html>
```

요청 파라미터 이름	요청 파라미터 값
reading	on
movie	on

6. 폼 데이터 처리하기

예제 6-6 [예제 6-4]의 회원 가입 양식에서 폼 데이터를 한 번에 전송받기

JSPBook/WebContent/ch06/form06.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 ... (생략) ...
03     <form action="form06_process.jsp" name="member" method="post">
04 ... (생략) ...
```

6. 폼 데이터 처리하기

JSPBook/WebContent/ch06/form06_process.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="java.io.*,java.util.*"%>
03 <html>
04 <head>
05 <title>Form Processing</title>
06 </head>
07 <body>
08     <table border="1">
09         <tr>
10             <th>요청 파라미터 이름</th>
11             <th>요청 파라미터 값</th>
12         </tr>
13         <%
14             request.setCharacterEncoding("UTF-8");
15
16             Enumeration paramNames = request.getParameterNames();
17             while (paramNames.hasMoreElements()) {
18                 String name = (String) paramNames.nextElement();
19                 out.print("<tr><td>" + name + " </td>\n");
20                 String paramValue = request.getParameter(name);
21                 out.println("<td> " + paramValue + "</td></tr>\n");
22             }
23         <%>
24     </table>
25 </body>
26 </html>
```

요청 파라미터 이름	요청 파라미터 값
id	admin
passwd	1234
name	홍길순
phone	010
phone2	9002
phone3	7370
sex	여성
hobby	독서
comment	안녕하세요. 홍길순 가입합니다.

원리를 알면 IT가 맛있다
IT COOKBOOK



7. [웹 쇼핑몰] 상품 등록 페이지

상품 등록

Home

상품 등록

상품 코드: P1237

상품명: Galay Note5

가격: 900000

상세 정보: 212 8*125 6*6 6m, Super AMOLED display, Octa-Core processor

제조사: 삼성

분류: 핸드폰

재고 수: 100

상태: ☒ 신규 제품 ☐ 중고 제품 ☐ 재생 제품

[등록](#)

상품 목록

Home

iPhone 6s
4.7-inch, 1334X750 Retina HD display, 8-megapixel iSight Camera
800000원
[상세 정보 >](#)

LG PC 그램
13.3-inch, IPS LED display, 5rd Generation Intel Core processors
1500000원
[상세 정보 >](#)

Galaxy Tab S
212.8*125.6*6.6mm, Super AMOLED display, Octa-Core processor
900000원
[상세 정보 >](#)

Galay Note5
212 8*125 6*6 6m, Super AMOLED display, Octa-Core processor
900000원
[상세 정보 >](#)

© WebMarket

7. [웹 쇼핑몰] 상품 등록 페이지

예제 6-7 상품 정보 등록 페이지 만들기

WebMarket/WebContent/addProduct.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/
    css/bootstrap.min.css">
05 <script src="./resources/js/validation.js"X/script>
06 <title>상품 등록</title>
07 </head>
08 <body>
09     <jsp:include page="menu.jsp" />
10     <div class="jumbotron">
11         <div class="container">
12             <h1 class="display-3">상품 등록</h1>
13         </div>
14     </div>
15     <div class="container">
```

7. [웹 쇼핑몰] 상품 등록 페이지

```
15    <div class="container">
16        <form name="newProduct" action="./processAddProduct.jsp" class="form-
          horizontal" method="post">
17            <div class="form-group row">
18                <label class="col-sm-2">상품 코드</label>
19                <div class="col-sm-3">
20                    <input type="text" name="productId" class="form-
                      control" >
21                </div>
22            </div>
23            <div class="form-group row">
24                <label class="col-sm-2">상품명</label>
25                <div class="col-sm-3">
26                    <input type="text" name="name" class="form-control" >
27                </div>
28            </div>
29            <div class="form-group row">
```


7. [웹 쇼핑몰] 상품 등록 페이지

```
30         <label class="col-sm-2">가격</label>
31         <div class="col-sm-3">
32             <input type="text" name="unitPrice" class="form-
               control" >
33         </div>
34     </div>
35     <div class="form-group row">
36         <label class="col-sm-2">상세 정보</label>
37         <div class="col-sm-5">
38             <textarea name="description" cols="50" rows="2"
               class="form-control"></textarea>
39         </div>
40     </div>
41     <div class="form-group row">
42         <label class="col-sm-2">제조사</label>
43         <div class="col-sm-3">
44             <input type="text" name="manufacturer" class="form-
               control">
45         </div>
46     </div>
```

7. [웹 쇼핑몰] 상품 등록 페이지

```
47         <div class="form-group row">
48             <label class="col-sm-2">분류</label>
49             <div class="col-sm-3">
50                 <input type="text" name="category" class="form-
                    control" >
51             </div>
52         </div>
53         <div class="form-group row">
54             <label class="col-sm-2">재고 수</label>
55             <div class="col-sm-3">
56                 <input type="text" name="unitsInStock" class="form-
                    control" >
57             </div>
58         </div>
59         <div class="form-group row">
60             <label class="col-sm-2">상태</label>
61             <div class="col-sm-5">
62                 <input type="radio" name="condition" value="New " >
                    신규 제품
```

7. [웹 쇼핑몰] 상품 등록 페이지

```
63         <input type="radio" name="condition" value="Old" >  
           중고 제품  
64         <input type="radio" name="condition" value=  
           "Refurbished" > 재생 제품  
65     </div>  
66 </div>  
67 <div class="form-group row">  
68     <div class="col-sm-offset-2 col-sm-10 ">  
69         <input type="submit" class="btn btn-primary" value=  
           "등록" >  
70     </div>  
71 </div>  
72 </form>  
73 </div>  
74 </body>  
75 </html>
```

7. [웹 쇼핑몰] 상품 등록 페이지

예제 6-8 상품 정보 등록 데이터 처리하기

- [상품 데이터 접근 클래스 만들기]
 - 신규 상품 데이터를 저장하는 메소드 만들기:

WebMarket/src/dao/ProductRepository.java

```
01 package dao;
02 ... (생략) ...
03 public class ProductRepository {
04
05     private ArrayList<Product> listOfProducts = new ArrayList<Product>();
06     private static ProductRepository instance = new ProductRepository();
07
08     public static ProductRepository getInstance(){
09         return instance;
10     }
11     ... (생략) ...
12     public void addProduct(Product product) {
13         listOfProducts.add(product);
14     }
15 }
```

7. [웹 쇼핑몰] 상품 등록 페이지

■ 신규 상품 등록 처리 페이지 만들기

WebMarket/WebContent/processAddProduct.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="dto.Product"%>
03 <%@ page import="dao.ProductRepository"%>
04
05 <%
06     request.setCharacterEncoding("UTF-8");
07
08     String productId = request.getParameter("productId");
09     String name = request.getParameter("name");
10     String unitPrice = request.getParameter("unitPrice");
11     String description = request.getParameter("description");
12     String manufacturer = request.getParameter("manufacturer");
13     String category = request.getParameter("category");
14     String unitsInStock = request.getParameter("unitsInStock");
15     String condition = request.getParameter("condition");
```

7. [웹 쇼핑몰] 상품 등록 페이지

```
16
17     Integer price;
18
19     if (unitPrice.isEmpty())
20         price = 0;
21     else
22         price = Integer.valueOf(unitPrice);
23
24     long stock;
25
26     if (unitsInStock.isEmpty())
27         stock = 0;
28     else
29         stock = Long.valueOf(unitsInStock);
30
```

7. [웹 쇼핑몰] 상품 등록 페이지

```
31     ProductRepository dao = ProductRepository.getInstance();
32
33     Product newProduct = new Product();
34     newProduct.setProductId(productId);
35     newProduct.setPname(name);
36     newProduct.setUnitPrice(price);
37     newProduct.setDescription(description);
38     newProduct.setManufacturer(manufacturer);
39     newProduct.setCategory(category);
40     newProduct.setUnitsInStock(stock);
41     newProduct.setCondition(condition);
42
43     dao.addProduct(newProduct);
44
45     response.sendRedirect("products.jsp");
46 %>
```

7. [웹 쇼핑몰] 상품 등록 페이지

■ 상품 목록 페이지 수정하기

WebMarket/WebContent/products.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="java.util.ArrayList"%>
03 <%@ page import="dto.Product"%>
04 <%@ page import="dao.ProductRepository"%>
05 ...(생략)...
06 <%
07     ProductRepository dao = ProductRepository.getInstance();
08     ArrayList<Product> listOfProducts = dao.getAllProducts();
09 %>
10 <div class="container">

11     <div class="row" align="center">
12 ...(생략)...
```


7. [웹 쇼핑몰] 상품 등록 페이지

- 상품 상세 정보 페이지 수정하기

WebMarket/WebContent/product.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="dto.Product"%>
03 <%@ page import="dao.ProductRepository"%>
04 ...(생략)...
05     <%
06         String id = request.getParameter("id");
07         ProductRepository dao = ProductRepository.getInstance();
08         Product product = dao.getProductById(id);
09     %>
10     <div class="container">
11         <div class="row">
12 ...(생략)...
```