

# JSP와 서블릿

## (Web Programming)

# 제1장 간단한 웹 프로그램 실행

## 1.1 JSP와 Servlet에 대하여

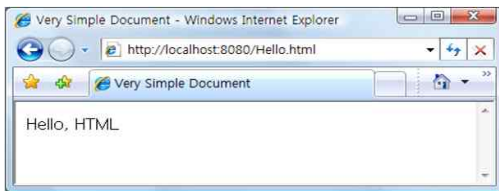
- 서블릿(servlet)이란 자바 클래스를 기반으로 하는 웹 애플리케이션 프로그래밍 기술이다.
- JSP(Java Server Page)도 자바를 기반으로 하는 웹 애플리케이션 기술로 HTML 문서 안에 자바 코드가 삽입되는 구조를 갖는다.

## 1.2 간단한 HTML문서로 톰캣 테스트하기

- 웹 서버 안에서 서블릿과 JSP를 지원하는 부분을 웹 컨테이너(Web Container)라 하고 톰캣은 웹 컨테이너를 갖춘 웹서버이다.
- root웹 애플리케이션 디렉터리에 있는 HTML 문서는 http://[웹 서버 주소]:[포트번호]/[파일이름] 포맷의 URL을 이용해 가져올 수 있다.
- 톰캣 포트 오류인 경우 (netstat tcp 옵션확인)

```
netstat -anofind "8088"  
taskkill /f /pid 18064
```

### [예제1-1] 아주 간단한 HTML 문서



[WebContent]-[ROOT]-[Hello.html]

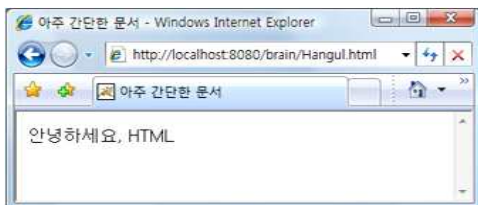
```
<html>  
  <head><title>Very Simple Document.</title></head>  
  <body>  
    Hello, HTML  
  </body>  
</html>
```

### [예제1-2] 한글 HTML 문서 [미완성]

[WebContent]-[brain]-[Hangul.html]

```
<html>  
  <head><title>아주 간단한 문서.</title></head>  
  <body>  
    안녕하세요, HTML  
  </body>  
</html>
```

### [예제1-3] 한글HTML문서 [완성]

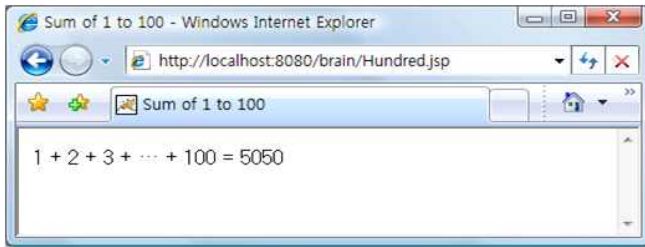


- http-equiv="Content-Type" : 이문서의 내용에 대한 설정이다.
- cotent="text/html" : HTML 문서로 작성된 텍스트이다.
- charset="euc-kr" : euc-kr 문자셋(한글코드)으로 인코딩 되어 있다.

[WebContent]-[brain]-[Hangul.html]

```
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=euc-kr">  
    <title>아주 간단한 문서</title>  
  </head>  
  <body>안녕하세요, HTML</body>  
</html>
```

#### [예제1-4] 간단한 JSP 페이지

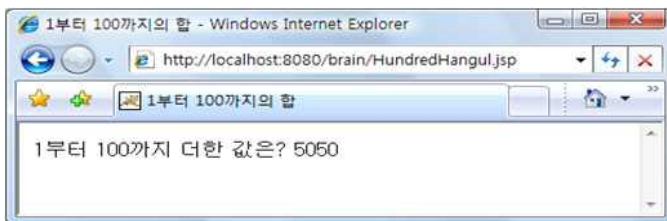


- 스크립틀릿(scriptlet): 작은 스크립트란 뜻으로 <%로 시작해서 %>로 끝나고, 그 사이에 자바 명령문들을 쓸 수 있다.
- 익스프레션(Expression): 식이란 뜻으로 <%=로 시작해서 %>로 끝나고, 그 사이에 자바식을 쓸 수 있다.
- JSP페이지에 있는 HTML 코드는 웹 브라우저로 그대로 전달되지만 스크립틀릿과 익스프레션은 웹 컨테이너에 의해 처리된다.
- 웹컨테이너는 포함된 자바 명령문은 실행만 하고, 익스프레션에 포함된 자바 식은 계산한 다음에 그 결과를 웹 브라우저로 보낸다.

[WebContent]-[brain]-[Hundred.jsp]

```
<html>
  <head><title>Sum of 1 to 100</title></head>
  <body>
    <%
      int total = 0;
      for(int cnt=1; cnt<=100; cnt++) {
        total+=cnt;
      }
    %>
    1 + 2 + 3 + ... + 100 = <%=total%>
  </body>
</html>
```

#### [예제1-5] 한글이 포함된 JSP 페이지



- 한글이 있는 JSP 페이지는 HTML 문서의 경우처럼 <META> 태그를 사용하면 안 되고 1행의 @page 코드처럼 지시자(directive)라고 하는 JSP 문법을 이용해 웹 컨테이너가 JSP 페이지를 처리할 때 필요한 정보를 기술해 처리한다.
- content-Type = "text/html; charset=euc-kr" : 이 문서의 내용은 HTML 문법으로 작성된 텍스트이고 euc-kr 문자셋 (한글코드)으로 인코딩되어져 있다.

[WebContent]-[brain]-[HundredHangul.jsp]

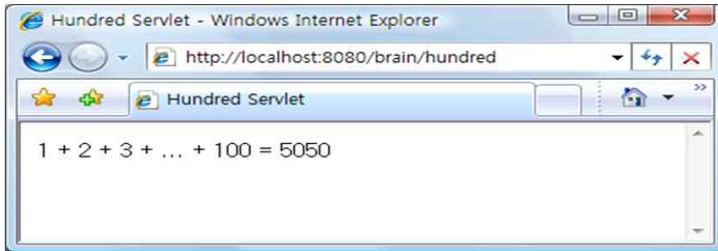
```
<%@page contentType="text/html; charset=euc-kr"%>
<html>
  <head><title>1부터 100까지의 합</title></head>
  <body>
    <%
      int total=0;
      for(int cnt=1; cnt<=100; cnt++){
        total+=cnt;
      }
    %>
    1부터 100까지 더한 값은? <%=total%>
    <%
      totla=0;
      for(int cnt=0; cnt<=100; cnt=cnt+2){
        total +=cnt;
      }
    %>
    1부터 100까지 짝수를 더한 값은? <%=total%>
  </body>
</html>
```

## 제2장 서블릿의 기초

### 2.1 서블릿 클래스 작성하기

- 웹컨테이너는 서블릿 클래스를 가지고 서블릿 객체를 만든 다음 그 객체를 초기화해서 웹 서비스를 할 수 있는 상태로 만드는데, 이 작업을 거친 서블릿 객체만 서블릿이라고 할 수 있다
- 서블릿 클래스를 작성할 때 지켜야할 세 가지 규칙
  - 1) 서블릿 클래스는 javax.servlet.http.HttpServlet 클래스를 상속하도록 만들어야한다.
  - 2) doGet 또는 doPost 메서드 안에 웹 브라우저로부터 요청이 왔을 때 해야 할 일을 기술해야한다
  - 3) HTML 문서는 doGet, doPost 메서드의 두 번째 파라미터를 이용해서 출력해야한다

[예제2-1] 1부터 100까지 더하는 서블릿 클래스



[brain]-[JavaResources]-[src]-[HundredServlet.java]

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
@WebServlet("/hundred")
public class HundredServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        int total = 0;

        for(int cnt=1; cnt<101; cnt++){
            total+=cnt;
        }
        PrintWriter out=response.getWriter();
        out.println("<html> ");
        out.println("<head><title>Hundred Servlet</title></head>");
        out.println("<body>");
        out.printf("1 + 2 + 3 + ... + 100 = %d", total);
        out.println("</body>");
        out.println("</html>");
    }
}
```

[예제2-2] 한글 출력을 포함한 1부터 100까지 더하는 서블릿 클래스

[brain]-[JavaResources]-[src]-[HundredServlet.java] 수정

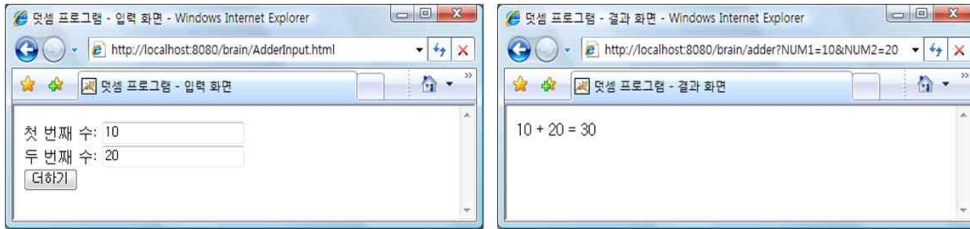
```
@WebServlet("/hundred")
public class HundredServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException{
        int total=0;

        for(int cnt=1; cnt<101; cnt++){
            total+=cnt;
        }
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        out.println("<html>");
        out.println("<head><title>1부터 100까지 더하는 서블릿</title></head>");
        out.println("<body>");
        out.printf("1부터 100까지의 합은 = %d", total);
        out.println("</body>");
        out.println("</html>");
    }
}
```

## 2.2 웹브라우저로부터 데이터를 입력받는 서블릿 클래스

- 웹 페이지를 통해 두 수를 입력 받은 후 그 둘을 합한 결과를 오른쪽 웹 페이지를 통해 보여주는 웹 애플리케이션이다.
- 둘 이상의 웹페이지로 구성되는 웹 어플리케이션을 개발할 때는 먼저 화면을 설계하고 다음에 각 화면 URL을 정하고 코딩하는 것이 좋다.

[예제2-3] 두 개의 수를 입력받는 HTML 문서



[brain]-[WebContent]-[AdderInput.html]

```
<html>
<head><title>덧셈 프로그램 - 입력 화면</title></head>
<body>
  <form action="/adder">
    첫 번째 수: <input type="text" name="num1"><br>
    두 번째 수: <input type="text" name="num2"><br>
    <input type="submit" value="더하기">
  </form>
</body>
</html>
```

[예제2-4] 두 수의 합을 구하는 서블릿 클래스

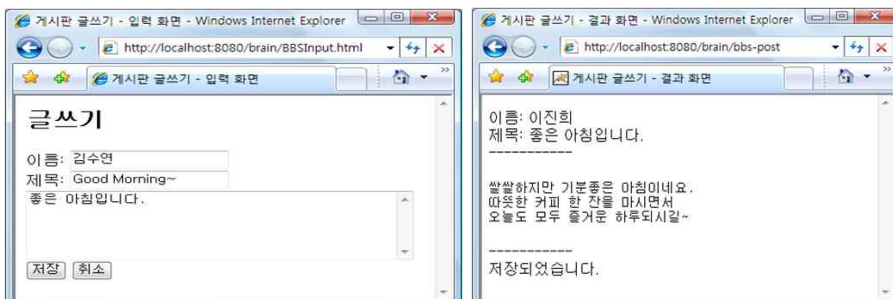
[brain]-[JavaResources]-[src]-[AdderServlet.java]

```
@WebServlet("/adder")
public class AdderServlet extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String str1=request.getParameter("num1");
        String str2=request.getParameter("num2");
        int num1=Integer.parseInt(str1);
        int num2=Integer.parseInt(str2);
        int sum=num1 + num2;
        out.println("<html>");
        out.println("<head><title>덧셈 프로그램 - 결과 화면</title></head>");
        out.println("<body>");
        out.printf("%d + %d = %d", num1, num2, sum);
        out.println("</body>");
        out.println("</html>");
    }
}
```

## 2.3 POST 메서드를 이용한 데이터 전송

- 입력 데이터가 클 경우에는 URL 뒷부분이 잘릴 수 있으므로 URL이 아닌 별도의 영역을 통해 입력 데이터를 전송해야 한다.
- <form> 태그에 method라는 애트리뷰트를 추가하고 값으로 post를 지정하면 웹서버는 입력데이터가 어디에 저장되어 있는지 판단한다.

[예제2-5] 게시판 글쓰기 기능의 데이터 입력을 위한 HTML 문서



[brain]-[WebContent]-[BBSInput.html]

```
<html>
<head><title>게시판 글쓰기 - 입력 화면</title></head>
<body>
  <h2>글쓰기</h2>
  <form action="/bbs-post" method="post">
    이름: <input type="text" name="name"><br>
    제목: <input type="text" name="title"><br>
    <textarea cols=50 rows=5 name="content"></textarea><br>
    <input type="submit" value="저장">
    <input type="reset" value="취소">
  </form>
</body>
</html>
```

#### [예제2-6] 게시판 글쓰기 기능을 처리하는 서블릿 클래스 [미완성]

[brain]-[JavaResources]-[src]-[BBSPostServlet]

```
@WebServlet("/bbs-post")
public class BBSPostServlet extends HttpServlet{
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        String title = request.getParameter("title");
        String content = request.getParameter("content");
        out.println("<html>");
        out.println("<head><title>게시판 글쓰기 - 결과 화면</title></head>");
        out.println("<body>");
        out.printf("이름: %s <br>", name);
        out.printf("제목: %s <br>", title);
        out.println("-----<br>");
        out.printf("<pre>%s</pre>", content);
        out.println("-----<br>");
        out.println("저장되었습니다.");
        out.println("</body>");
        out.println("</html>");
    }
}
```

#### [예제2-7] 게시판 글쓰기 기능을 처리하는 서블릿 클래스 [완성]

[brain]-[JavaResources]-[src]-[BBSPostServlet]

```
@WebServlet("/bbs-post")
public class BBSPostServlet extends HttpServlet{
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException{
        response.setContentType("text/html;charset=UTF-8");
        request.setCharacterEncoding("euc-kr");
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        String title = request.getParameter("title");
        String content = request.getParameter("content");
        out.println("<html>");
        out.println("<head><title>게시판 글쓰기 - 결과 화면</title></head>");
        out.println("<body>");
        out.printf("이름: %s <br>", name);
        out.printf("제목: %s <br>", title);
        out.println("-----<br>");
        out.printf("<pre>%s</pre>", content);
        out.println("-----<br>");
        out.println("저장되었습니다.");
        out.println("</body>");
        out.println("</html>");
    }
}
```

## 2.4 다양한 형태로 데이터 입력 받기

- <form> 태그를 이용하면 텍스트 상자 외에도 라디오 버튼 체크 박스, 선택상자 등 다양한 형태로 데이터를 입력 받을 수 있다.
- 패스워드: <input> 태그의 password 속성값을 지정해서 만든다. 입력한 문자는 모니터 상에 나타나지 않는다.

```
<input type="text" name="name">
<input type="text" name="id">
<input type="password" name="password">
```



- 라디오버튼: <input>태그의 radio 속성값을 지정해서 만들 수 있으며 반드시 name, value 속성을 써야 한다.

```
<input type="radio" name="gender" value="male">
<input type="radio" name="gender" value="female">
```



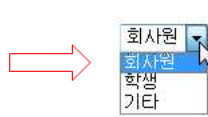
- 체크 박스: <input> 태그의 checkbox 속성값을 지정해서 만들 수 있으며 name 속성은 다른 값을 지정해야 한다.

```
<input type="checkbox" name="iNotice">
<input type="checkbox" name="cNotice">
<input type="checkbox" name="dNotice">
```

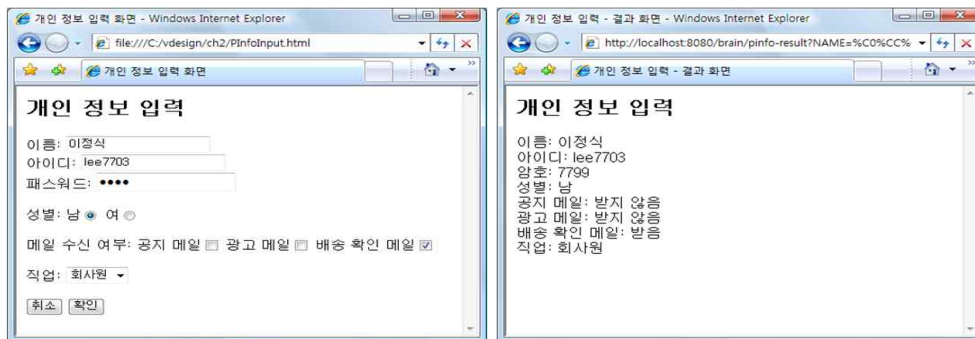


- 선택 상자: <select> 태그를 이용해서 만들 수 있고 시작 태그와 끝 태그 사이에 선택 항목의 이름을 포함한 <option> 태그들을 써야 한다.

```
<select name="job">
  <option>회사원</option>
  <option>학생</option>
  <option>기타</option>
</select>
```



### [예제2-8] 개인 정보를 입력받는 HTML 문서



[brain]-[WebContent]-[PInfoInput.html]

```
<html>
<head><title>개인 정보 입력 화면</title></head>
<body>
  <h2>개인 정보 입력</h2>
  <form action="/pinfo-result" method="get">
    이름: <input type="text" name="name"><br>
    아이디: <input type="text" name="id"><br>
    패스워드: <input type="password" name="password"><br><br>
    성별: 남 <input type="radio" name="gender" value="male">
    여 <input type="radio" name="gender" value="female"><br><br>
    메일 수신 여부:
      공지 메일 <input type="checkbox" name="iNotice">
      광고 메일 <input type="checkbox" name="cNotice">
      배송 확인 메일 <input type="checkbox" name="dNotice"><br><br>
    직업:
    <select name="job">
      <option>회사원</option>
      <option>학생</option>
      <option>기타</option>
    </select><br><br>
    <input type="reset" value="취소">
    <input type="submit" value="확인">
  </form>
</body>
</html>
```

[예제2-9] 개인 정보를 저장하는 서블릿 클래스

[brain]-[JavaResources]-[src]-[PersonalInfoServlet.java]

```
@WebServlet("/pinfo-result")
public class PersonalInfoServlet extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        String id = request.getParameter("id");
        String password = request.getParameter("password");
        String gender = request.getParameter("gender");
        if(gender.equals("male")){
            gender = "남";
        }else{
            gender = "여";
        }
        String iNotice = request.getParameter("iNotice");
        String cNotice = request.getParameter("cNotice");
        String dNotice = request.getParameter("dNotice");
        String job = request.getParameter("job");
        out.println("<html>");
        out.println("<head><title>개인 정보 입력 - 결과 화면</title></head>");
        out.println("<body>");
        out.println("<h2>개인 정보 입력</h2>");
        out.printf("이름: %s <br>", name);
        out.printf("아이디: %s <br>", id);
        out.printf("암호: %s <br>", password);
        out.printf("성별: %s <br>", gender);
        out.printf("공지 메일: %s <br>", noticeToHangul(iNotice));
        out.printf("광고 메일: %s <br>", noticeToHangul(cNotice));
        out.printf("배송 확인 메일: %s <br>", noticeToHangul(dNotice));
        out.printf("직업: %s <BR>", job);
        out.println("</body>");
        out.println("</html>");
    }
    private String noticeToHangul(String notice){
        if(notice == null){
            return "받지 않음";
        }else if(notice.equals("on")){
            return "받음";
        }else{
            return notice;
        }
    }
}
```

<form> 태그의 get 메서드로 한글 입력을 위해서는 server.xml 파일에 URLEncoder 속성을 추가하고 값으로 "euc-kr"를 지정한다.

[톰캣서버]-[server.xml] 추가

```
<connector port="8080" protocol="HTTP/1.1" URLEncoder="euc-kr" connection Timeout="20000"/>
```



## 제3장 JSP의 기초

### 3.1 스크립팅 요소의 문법

- 스크립틀릿(scriptlet): 스크립틀릿은 <%로 시작해서 %>로 끝나고 그 사이 자바 명령문이 들어가고 웹 서버 쪽에서 실행된다.
- 익스프레션(expression): 익스프레션은 <%=로 시작해서 %>로 끝나고 그 사이에 자바 식, 연산자, 메서드 호출식이 들어갈 수 있다.
- 선언부(declaration): <%!로 시작해서 %>로 끝나고 그사이에 변수 선언문이나 메서드 선언문을 쓸 수 있다.

[예제3-1] 여러 개의 스크립틀릿이 있는 JSP 페이지

```
[WebContent]-[TwoHundred.jsp]

<html>
  <head><title>1부터 100까지의 합</title></head>
  <body>
    <%
      int total = 0;
      for (int cnt = 1; cnt <= 100; cnt++){
        total += cnt;
      }
    %>
    1부터 100까지의 합 = <%=total%> <br>
  </body>
</html>
```

[예제3-2] 선언부를 포함한 JSP페이지

```
[WebContent]-[TwoPower.jsp]

<html>
  <head><title>2의 거듭제곱</title></head>
  <body>
    2 ^ 1 = <%=power(2, 1)%> <br>
    2 ^ 2 = <%=power(2, 2)%> <br>
    2 ^ 3 = <%=power(2, 3)%> <br>
  </body>
</html>
<%!
  private int power(int base, int exponent){
    int result= 1;
    for (int cnt = 0; cnt < exponent; cnt++){
      result *= base;
    }
    return result;
  }
%>
```

### 3.2 지시자의 문법

- 웹 브라우저로부터의 요청을 처리하는 것이 아니라 웹컨테이너가 JSP 페이지를 서블릿 클래스로 변환할 때 필요한 정보들을 기술한다.
- JSP 페이지에 사용할 수 있는 지시자는 page 지시자, include 지시자, taglib 지시자등이 있다.

[예제3-3] page 지시자의 import 애트리뷰트 사용 예

```
[WebContent]-[DateTime.jsp]

<%@page import="java.util.GregorianCalendar"%>
<html>
  <head><title>날짜와 시각</title></head>
  <body>
    <%
      GregorianCalendar now = new GregorianCalendar();
      String date = String.format("%TF", now);
      String time = String.format("%TT", now);
    %>
    오늘의 날짜: <%=date%> <br> 현재의 시각: <%=time%> <br>
  </body>
</html>
```

#### [예제3-4] include 지시자의 사용 예

[WebContent]-[Menu.jsp]

```
<html>
  <head><title>오늘의 메뉴</title></head>
  <body>
    <h3>오늘의 메뉴</h3>
    - 삼계탕 <br>
    - 돈까스 <br>
    - 튀김국수 <br><br>
    <%@include file="Today.jsp"%>
  </body>
</html>
```

[WebContent]-[Today.jsp]

```
<%@page import="java.util.*"%>
<%GregorianCalendar now=new GregorianCalendar();%>
<%=String.format("%TY년 %Tm월 %Td일", now, now, now)%>
```

### 3.3 주석을 기술하는 방법

- JSP 페이지의 HTML 코드부분: <!--로 시작해서 -->로 끝나는 HTML주석을 쓸 수 있다.
- JSP 페이지의 스크립팅 요소 안: 자바 문법을 따르는 주석을 쓸 수 있다.

#### [예제3-5] 여러 가지 주석을 포함하는 JSP 페이지

[WebContent]-[TenMultiply.jsp]

```
<html>
  <head><title>1부터 10까지의 곱</title></head>
  <!-- 이것은 JSP에 의해 생성된 HTML 문서입니다. -->
  <body>
    <%
      int result = 1; // 곱을 저장할 변수

      for (int cnt = 1; cnt <= 10; cnt++) { //1부터 10까지 곱하는 반복문
        result *= cnt;
      }
    %>
    1부터 10까지 곱한 값은? <%=result%>
  </body>
</html>
```

### 3.4 request 내장 변수

- JSP 페이지의 내장 변수: JSP 페이지 안에 선언을 하지 않고도 사용할 수 있는 변수 이다.
- 웹 컨테이너가 JSP 페이지를 서블릿 클래스로 변환할 때 자동으로 내장 변수를 선언하므로 JSP에서 내장 변수를 사용할 수 있다.
- request 내장 변수는 서블릿 클래스의 doGet, doPost 메서드의 첫 번째 파라미터와 동일한 역할을 한다.
- 둘이상의 화면으로 구성된 애플리케이션은 먼저 URL을 정한 뒤 각 URL에 해당하는 코드를 작성하는 것이 좋다.

1)http://localhost:8080/YourName.jsp: 웹 브라우저로부터 이름을 입력받는 HTML 문서

2)http://localhost:8080/Hi.jsp: 입력된 이름을 가지고 인사말을 출력하는 JSP 문서

#### [예제3-6] 웹 브라우저로부터 이름을 입력받는 HTML 문서

[WebContent]-[YourName.jsp]

```
<html>
  <head><title>이름 입력</title></head>
  <body>
    당신의 이름을 입력하세요.
    <form action="Hi.jsp" method="get">
      이름: <input type="text" name="yourname">
      <input type="submit" value="확인">
    </form>
  </body>
</html>
```

[예제3-7] 입력된 이름을 가지고 인사말을 출력하는 JSP 페이지

[WebContent]-[Hi.jsp]

```
<html>
  <head><title>인사하기</title></head>
  <body>
    안녕하세요, <%=request.getParameter("yourname")%>님
  </body>
</html>
```

### 3.5 out 내장 변수

- JSP 페이지는 HTML 코드와 익스프레션만으로 HTML 문서를 만들어서 출력할 수 있으므로 서블릿처럼 print 메서드를 호출할 필요가 없다.
- out 내장 변수는 서블릿 클래스에서 getWriter 메서드를 호출해서 얻은 PrintWriter 객체와 비슷한 역할을 한다.

[예제3-8] 1부터 10까지 출력하는 JSP페이지

[WebContent]-[OneToTen.jsp]

```
<html>
  <head><title>1부터 10까지 순서대로</title></head>
  <body>
    <h3>1부터 10까지 순서대로</h3>
    <% for(int cnt = 1; cnt <= 10; cnt++) { %>
      <% =cnt %><br>
    <% } %>
  </body>
</html>
```

[예제3-9] out 내장 변수를 사용하는 JSP페이지

[WebContent]-[OneToTen.jsp]

```
<html>
  <head><title>1부터 10까지 순서대로</title></head>
  <body>
    <h3>1부터 10까지 순서대로</h3>
    <%
      for(int cnt = 1; cnt <= 10; cnt++) {
        out.println(cnt + "<br>");
      }
    %>
  </body>
</html>
```

[예제3-10] 출력 버퍼의 크기를 바꾸는 JSP 페이지

[WebContent]-[BufferSize.jsp]

```
<%@page buffer= "4kb"%>    //버퍼를 거치지 않고 웹브라우저로 바로 전송하려면 buffer 애틀리뷰트에 'none'값을 저장한다.
<html>
  <head><title>출력 버퍼의 크기 지정</title></head>
  <body>
    출력 버퍼의 크기 : <%=out.getBufferSize()%> 바이트
  </body>
</html>
```

### 3.6 response 내장 변수

- response 내장 변수는 서블릿 클래스에 있는 doGet, doPost 메서드의 두 번째 파라미터와 동일한 역할을 한다.
- sendRedirect 메서드를 호출할 때 주의할 점: 이 메서드를 호출하기 전과 후에 웹 브라우저 데이터를 출력하면 안 된다.

[예제3-11] 다른 웹 페이지로 이동하는 JSP 페이지

[WebContent]-[Move.jsp]

```
<%response.sendRedirect("http://www.hanb.co.kr/");%>
```

### 3.7 Application 내장 변수

- application 내장 변수의 getContextPath 메서드는 웹 애플리케이션의 URL 경로명을 리턴 한다.
- application 내장 변수의 getRealPath 메서드는 웹 애플리케이션 내에서 파일 경로명을 파일시스템 전체에 대한 절대 경로명을 리턴 한다.

[예제3-12] application 내장 변수의 사용 예

[WebContent]-[ApplicationTest.jsp]

```
<html>
  <head><title>application 내장 변수의 사용 예</title></head>
  <body>
    <%
      String appPath = application.getContextPath();
      String filePath = application.getRealPath("/Intro.html");
    %>
    웹 애플리케이션의 URL 경로명: <%=appPath%> <BR>
    /Intro.html의 파일 경로명: <%=filePath%> <BR>
  </body>
</html>
```

### 3.8 파일로부터 데이터를 읽는 방법

- JSP 페이지 안에서 파일을 읽기 위해서는 기본적으로 파일의 절대 경로를 사용하거나 톰캣의 설치 디렉터리로부터 상대 경로명을 사용해야 한다.
- 경로명을 사용하는 가장 좋은 방법은 getRealPath 메서드를 이용해서 애플리케이션 내에서의 경로명을 절대 경로명으로 바꾸어 사용한다.

[예제3-13] 텍스트 파일을 읽는 스탠드얼론 프로그램

[JavaResources]-[src]-[FileReadProgram.java]

```
class FileReadProgram{
  public static void main(String args[]){
    BufferedReader reader = null;
    try{
      reader = new BufferedReader(new FileReader("c:/Temp/input.txt"));

      while(true){
        String str = reader.readLine();
        if (str == null){ break; }
        System.out.println(str);
      }
      reader.close();
    }catch(Exception e){ System.out.println("파일을 읽을 수 없습니다."); }
  }
}
```

[c:/Temp/input.txt]

내 귀는 소라껍질  
바다 소리를 그리워한다.

[예제3-14] 텍스트 파일의 내용을 읽어서 웹 브라우저로 출력하는 JSP 페이지

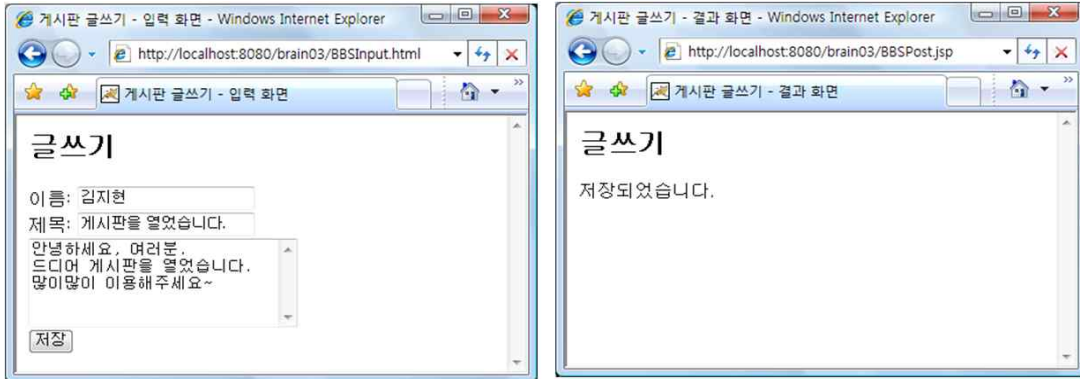
[WebContent]-[FileReader.jsp]

```
<%@page import="java.io.*"%>
<html>
  <head><title> 파일 읽기</title></head>
  <body>
    <%
      BufferedReader reader = null;
      try{
        String filePath = application.getRealPath("/WEB-INF/input.txt");
        reader = new BufferedReader(new FileReader(filePath));

        while(true){
          String str = reader.readLine();
          if(str == null) { break; }
          out.println(str + "<br>");
        }
        reader.close();
      }catch(Exception e) { out.println("파일을 읽을 수 없습니다."); }
    %>
  </body>
</html>
```

### 3.9 파일에 데이터를 쓰는 방법

JSP 페이지에서도 `java.io.FileWriter`, `java.io.PrintWriter`, `java.io.FileOutputStream` 클래스 등을 이용한다.



[예제3-15] 게시판 글쓰기 기능의 입력 화면을 구현하는 HTML 문서

[WebContent]-[BBSInput.html]

```
<html>
  <body>
    <h2>글쓰기</h2>
    <form action="BBSPost.jsp" method="post">
      이름: <input type="text" name="name"><br>
      제목: <input type="text" name="title"><br>
      <textarea cols=30 rows=5 name="content"></textarea><br>
      <input type="submit" value="저장">
    </form>
  </body>
</html>
```

[예제3-16] 게시글을 저장하는JSP 페이지

[WebContent]-[BBSPost.jsp]

```
<%@page import="java.io.*, java.util.Date"%>
<html>
  <head><title>게시판 글쓰기 - 결과 화면</title></head>
  <body>
    <h2>글쓰기</h2>
    <%
      request.setCharacterEncoding("euc-kr");
      String name = request.getParameter("name");
      String title = request.getParameter("title");
      String content = request.getParameter("content");
      Date date = new Date();
      Long time = date.getTime();
      String fileName = time + ".txt";
      PrintWriter writer = null;
      try{
        String filePath = application.getRealPath("/WEB-INF/" + fileName);
        writer = new PrintWriter(filePath);
        writer.printf("제목: %s %n", title);
        writer.printf("글쓴이: %s %n", name);
        writer.println(content);
        out.println("저장되었습니다."); //out.println(filePath); 파일Path를 출력하여 파일이 생성되었는지 확인한다.
        writer.close();
      }catch(Exception e){
        out.println("파일에 데이터를 쓸 수 없습니다.");
      }
    %>
  </body>
</html>
```

위 결과를 실행한 후 '새로고침' 버튼을 누르면 누를 때마다 새로운 파일이 생성된다. 이 문제를 해결하기 위해 글을 저장하는 JSP와 출력하는 JSP를 나누어 작성한 후 `sendRedirect` 메서드를 이용해서 전자가 후자를 호출하도록 만든다.

[예제 3-17] 게시글을 저장하는 JSP 페이지와 그 결과를 출력하는 JSP 페이지

[WebContent]-[BBSPost.jsp]

```
<%@page import="java.io.*, java.util.Date"%>
<%
    request.setCharacterEncoding("euc-kr");
    String name = request.getParameter("name");
    String title = request.getParameter("title");
    String content = request.getParameter("content");

    Date date = new Date();
    Long time = date.getTime();
    String fileName = time + ".txt";
    String result;
    PrintWriter writer = null;
    try{
        String filePath = application.getRealPath("/WEB-INF/" + fileName);
        writer.printf("제목: %s %n", title);
        writer.printf("글쓴이: %s %n", name);
        writer.println(content);
        result = "SUCCESS";
        writer.close();

    }catch(Exception e){
        result = "FAIL";
        out.println("파일에 데이터를 쓸 수 없습니다.");
    }
    response.sendRedirect("BBSPostResult.jsp?result=" + result);
%>
```

[WebContent]-[BBSPostResult.jsp]

```
<html>
<head><title>게시판 글쓰기- 결과화면</title></head>
<body>
    <h2>글쓰기</h2>
    <%
        String str=request.getParameter("result");
        if(str.equals("SUCCESS")){
            out.println("저장되었습니다.");
        }else{
            out.println("파일에 데이터를 쓸 수 없습니다.");
        }
    %>
</body>
</html>
```

### 3.10 forward 메서드의 사용 방법

- forward 메서드는 JSP 페이지 안에서 다른 JSP 페이지를 호출할 때 사용하는 메서드로 RequestDispatcher 객체로 호출할 수 있다.
- forward 메서드를 호출할 때에는 request 내장 변수와 response 내장 변수를 파라미터로 넘겨줘야 한다.
- 호출하는 JSP 페이지로 데이터를 넘겨주려면 먼저 request.setAttribute 메서드로 request 내장 변수 안에 데이터를 저장해 놓아야 한다.
- 호출된 JSP 페이지 안에서 request 내장 변수 안의 데이터를 가져오려면 request.getAttribute 메서드를 호출하면 된다.

[예제3-18] 두 개의 수를 입력받는 HTML 문서

[WebContent]-[FourRulesInput.html]

```
<html>
<head><title>두 개의 수를 입력</title></head>
<body>
    <form action="FourRules.jsp">
        첫 번째 수: <input type="text" name="num1"><br>
        두 번째 수: <input type="text" name="num2"><br>
        <input type="submit" value="입력">
    </form>
</body>
</html>
```

[예제3-19] 사칙 연산을 수행하는 JSP 페이지와 그 결과를 출력하는 JSP 페이지

[WebContent]-[FourRules.jsp]

```
<%
String str1=request.getParameter("num1");
String str2=request.getParameter("num2");
int num1=Integer.parseInt(str1);
int num2=Integer.parseInt(str2);

request.setAttribute("sum", num1 + num2);
request.setAttribute("difference", num1-num2);
request.setAttribute("product",num1*num2);
request.setAttribute("quotient", num1/num2);

RequestDispatcher dispatcher=request.getRequestDispatcher("FourRulesResult.jsp");
dispatcher.forward(request, response);
%>
```

[WebContent]-[FourRulesResult.jsp]

```
<html>
<head><title>사칙연산</title></head>
<body>
    덧셈의 결과는<%=request.getAttribute("sum")%><br>
    뺄셈의 결과는<%=request.getAttribute("difference")%><br>
    곱셈의 결과는<%=request.getAttribute("product")%><br>
    나눗셈의 결과는<%=request.getAttribute("quotient")%><br>
</body>
</html>
```

### 3.11 include 메서드의 사용 방법

include 메서드도 다른 JSP 페이지를 호출하지만 호출된 JSP 페이지가 끝나고 나면 실행 흐름의 제어가 본래의 JSP페이지로 돌아온다.

[예제3-20] include 메서드의 사용 예

[WebContent]-[ChinesMenu.jsp]

```
<html>
<head><title>오늘의 메뉴</title></head>
<body>
    <h3>오늘의 메뉴</h3>
    - 짜장면 <br>
    - 볶음밥 <br>
    - 짬뽕 <br><br>
<%
    out.flush(); //HTML 코드가 올바른 순서로 출력되도록 out.flush() 메서드를 호출해서 출력 버퍼의 내용을 웹 브라우저로 보낸다.
    RequestDispatcher dispatcher = request.getRequestDispatcher("Now.jsp");
    dispatcher.include(request, response);
%>
</body>
</html>
```

[WebContent]-[Now.jsp]

```
<%@page import="java.util.*"%>
<%GregorianCalendar now = new GregorianCalendar();%>
[현재의 시각] <%=String.format("%TF %TT", now, now)%>
```

## 제4장 쿠키와 세션

### 4.1 쿠키기술의 사용방법

- 쿠키 기술은 전달할 데이터를 웹 브라우저로 보냈다가 웹 서버 쪽으로 다시 되돌려 받는 방법을 사용한다.
- 첫 번째 웹 컴포넌트(JSP페이지와 서블릿 클래스)가 웹 브라우저로 HTML 문서를 보낼 때 전달할 데이터를 함께 보내며 웹 브라우저는 그 데이터를 저장해 두었다가 두 번째 웹 컴포넌트를 호출할 때 URL과 함께 웹 서버로 보낸다.

[예제4-1] 쿠키 데이터를 저장하는 JSP 페이지

쿠키 데이터를 웹브라우저 쪽에 저장하기 위해서는 Cookie 클래스의 객체를 만든 후 addCookie 메서드를 호출한다.

[WebContent]-[StoreCookies.jsp]

```
<% Cookie[] cookies = request.getCookies(); %>
<%
    response.addCookie(new Cookie("name", "John"));
    response.addCookie(new Cookie("gender", "Male"));
    response.addCookie(new Cookie("age", "15"));
%>
<html>
    <head><title>쿠키 데이터 저장하기</title></head>
    <body>
        쿠키 데이터가 저장되었습니다.
    </body>
</html>
```

[예제4-2] 쿠키 데이터를 읽는 JSP 페이지

getCookies 메서드는 웹 브라우저가 보낸 모든 쿠키를 Cookie 배열로 리턴하기 때문에 Cookie 배열 변수에 리턴값을 받아야 한다.

[WebContent]-[ReadCookies.jsp]

```
<html>
    <head><title>쿠키 데이터 읽기</title></head>
    <body>
        <%
            String name="";
            String gender="";
            String age="";

            for(Cookie cookie:request.getCookies()) {
                if(cookie.getName().equals("name")) {
                    name=cookie.getValue();
                }else if(cookie.getName().equals("gender")) {
                    gender=cookie.getValue();
                }else if(cookie.getName().equals("age")) {
                    age=cookie.getValue();
                }
            }
        %>
        이름:<%=name%><br>
        성별:<%=gender%><br>
        나이:<%=age%><br>
    </body>
</html>
```

[예제4-3] 쿠키 데이터의 값을 수정하는 JSP 페이지

쿠키값 수정은 저장방식과 같으며 addCookie하면 웹브라우저는 쿠키 중에 같은 이름의 것이 있으면 그것을 지우고 새로운 쿠키를 저장한다.

[WebContent]-[ModifyCookies.jsp]

```
<%response.addCookie(new Cookie("age", "16"));%>
<html>
    <head><title>쿠키 데이터 수정하기</title></head>
    <body>
        AGE 쿠키에 새로운 값이 저장되었습니다.
    </body>
</html>
```



[예제4-4] 쿠키를 삭제하는 JSP 페이지

쿠키 데이터를 삭제하기 위해서는 setMaxAge 메서드로 쿠키의 남은 수명을 0으로 설정한 후 addCookie 메서드를 호출한다.

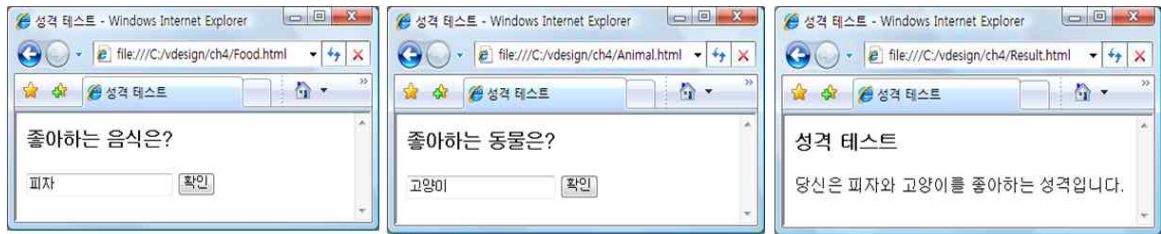
[WebContent]-[DeleteCookie.jsp]

```
<%
    Cookie cookie = new Cookie("gender", "");
    cookie.setMaxAge(0); //1시간 후에 쿠키가 지워지도록 만들려면 cookie.setMaxAge(3600) 초단위로 입력해준다.
    response.addCookie(cookie);
%>
<html>
    <head><title>쿠키 삭제하기</title></head>
    <body>
        GENDER 쿠키가 삭제되었습니다.
    </body>
</html>
```

4.2 세션 기술의 사용방법

세션 기술은 웹 브라우저를 거치지 않고 웹 서버에 있는 데이터 영역을 통해 데이터를 전달하는 방법이다.

[예제4-5] 성격 테스트의 첫 번째 화면을 출력하는 HTML 문서



[WebContent]-[pets]-[Food.html]

```
<html>
    <head><title>성격 테스트</title></head>
    <body>
        <h3>좋아하는 음식은?</h3>
        <form action="animal">
            <input type="text" name="food">
            <input type="submit" value="확인">
        </form>
    </body>
</html>
```

[예제4-10] 성격 테스트의 두 번째 화면을 출력하는 서블릿 클래스

[Java Resources]-[src]-[AnimalServlet.java]

```
@WebServlet("/animal")
public class AnimalServlet extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse response)throws IOException, ServletException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String food = request.getParameter("food");
        HttpSession session = request.getSession();
        session.setAttribute("food", food);
        out.println("<html>");
        out.println("<head><title>성격 테스트</title></head>");
        out.println("<body>");
        out.println("<h3>좋아하는 동물은?</h3>");
        out.println("<form action='result'>");
        out.println("<input type='text' name='animal'>");
        out.println("<input type='submit' value= '확인'>");
        out.println("</form>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

[예제4-11] 성격 테스트의 결과를 출력하는 서블릿 클래스

[Java Resources]-[src]-[ResultServlet.java]

```
@WebServlet("/result")
public class ResultServlet extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse response)throws IOException, ServletException{
        response.setContentType("text/html;charset=UTF-8");

        PrintWriter out = response.getWriter();

        HttpSession session = request.getSession();
        String food = (String)session.getAttribute("food");
        String animal = request.getParameter("animal");

        session.invalidate(); //세션을 종료한다.

        out.println("<head><title>성격 테스트</title></head>");
        out.println("<body>");
        out.println("<h3>성격 테스트</h3>");
        out.printf("당신은 %s와 %s를 좋아하는 성격입니다.", food, animal);
        out.println("</body>");
        out.println("</html>");
    }
}
```

### 4.3 JSP 페이지에서 세션 기술을 사용하는 방법

JSP 페이지에서는 JSP 페이지가 서블릿 클래스로 변환될 때 getSession 메서드가 자동 추가되기 때문에 이 메서드를 호출 할 필요가 없다.

[예제4-12] 개인 정보 입력 화면을 제공하는 HTML 문서

[WebContent]-[PersonalInfo.html]

```
<html>
<head>
    <title>회원 가입</title>
</head>
<body>
    개인정보를 입력하세요.

    <form action="Agreement.jsp" method="post">
        아이디:
        <input type="text" name="id"><br>
        비밀번호:
        <input type="password" name="password"><br>
        이름:
        <input type="text" name="name"><br>

        <input type="submit" value="확인">
    </form>
</body>
</html>
```

[예제4-13] 약관 동의 화면을 제공하는 JSP 페이지

[WebContent]-[Agreement.jsp]

```
<%
    request.setCharacterEncoding("euc-kr");

    String id = request.getParameter("id");
    String password = request.getParameter("password");
    String name = request.getParameter("name");

    session.setAttribute("id", id);
    session.setAttribute("password", password);
    session.setAttribute("name", name);
%>
```

[WebContent]-[Agreement.jsp]

```
<html>
  <head>
    <title>회원 가입</title>
  </head>
  <body>
    <h3>약관</h3>
    ----- <br>
    1. 회원 정보는 웹 사이트의 운영을 위해서만 사용됩니다. <br>
    2. 웹 사이트의 정상 운영을 방해하는 회원은 탈퇴 처리합니다. <br>
    ----- <br>
    <form action="Subscribe.jsp" method="post">
      위의 약관에 동의하십니까?
      <input type="radio" name="agree" value="yes">동의함
      <input type="radio" name="agree" value="no">동의하지 않음 <br><br>
      <input type="submit" value="확인">
    </form>
  </body>
</html>
```

[예제4-14] 회원 정보를 저장하는 JSP 페이지

[WebContent]-[Subscribe.jsp]

```
<%@page import="java.io.*"%>
<%
  String agree = request.getParameter("agree");
  String result = null;

  if(agree.equals("yes")) {
    String id = (String)session.getAttribute("id");
    String password = (String)session.getAttribute("password");
    String name = (String)session.getAttribute("name");
    PrintWriter writer = null;
    try {
      String filePath = application.getRealPath("/WEB-INF/" + id + ".txt");
      writer = new PrintWriter(filePath);
      writer.println("아이디:" + id);
      writer.println("패스워드:" + password);
      writer.println("이름:" + name);
      result = "SUCCESS";
      writer.close();
    }catch(Exception e) {
      result = "FAIL";
    }
  }
  session.invalidate(); //세션을 종료한다.
  response.sendRedirect("Result.jsp?result=" + result);
%>
```

[예제4-15] 회원 가입 결과를 출력하는 JSP 페이지

[WebContent]-[Result.jsp]

```
<% String result=request.getParameter("result");%>
<html>
  <head><title>회원 가입</title></head>
  <body>
    <h3>회원 가입 결과</h3>
    <%
      if(result.equals("SUCCESS"))
        out.println("가입되었습니다.");
      else
        out.println("가입되지 않았습니다.");
    %>
  </body>
</html>
```

## 제5장 익셉션 처리

### 5.1 웹 컴포넌트에서의 익셉션 처리방법

익셉션 처리를 위해서 프로그램에서 try~catch 문을 이용해서 익셉션을 처리한다.

[예제5-1] 두 수를 더하는 서블릿 클래스

[Java Resources]-[src]-[AdderServlet.java]

```
@WebServlet("/adder")
public class AdderServlet extends HttpServlet{
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();

        String str1=request.getParameter("num1");
        String str2=request.getParameter("num2");

        try {
            int num1=Integer.parseInt(str1);
            int num2=Integer.parseInt(str2);
            int result=num1 + num2;
            out.println("<html>");
            out.println("<head><title>덧셈프로그램</title></head>");
            out.println("<body>");
            out.printf("%d + %d = %d", num1, num2, result);
            out.println("</body>");
            out.println("</html>");

        }catch(NumberFormatException e) {
            out.println("<html>");
            out.println("<head><title>덧셈프로그램-에러화면</title></head>");
            out.println("<body>");
            out.println("잘못된 데이터가 입력되었습니다.");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

[예제5-2] 두 수를 더하는 JSP 페이지

[WebContent]-[Adder.jsp]

```
<%
    try{
        String str1=request.getParameter("num1");
        String str2=request.getParameter("num2");
        int num1=Integer.parseInt(str1);
        int num2=Integer.parseInt(str2);
        int result=num1 + num2;
    %>
    <html>
        <head><title>덧셈프로그램</title></head>
        <body>
            <%=num1%> + <%=num2%> = <%=result%>
        </body>
    </html>
<%
} catch(NumberFormatException e){
    %>
    <html>
        <head><title>덧셈프로그램 - 에러화면</title></head>
        <body>
            잘못된 데이터가 입력되었습니다.
        </body>
    </html>
<%
}
%>
```

### [예제5-3] 두 수를 더하는 JSP 페이지

[WebContent]-[Adder.jsp]

```
<%
    int num1=0, num2=0, result=0;
    try{
        String str1=request.getParameter("num1");
        String str2=request.getParameter("num2");
        num1=Integer.parseInt(str1);
        num2=Integer.parseInt(str2);
        result=num1 + num2;
    }catch(NumberFormatException e){
        RequestDispatcher dispatcher=request.getRequestDispatcher("DataError.jsp");
        dispatcher.forward(request, response);
    }
%>
<html>
    <head><title>덧셈프로그램</title></head>
    <body>
        <%=num1%> + <%=num2%> = <%=result%>
    </body>
</html>
```

### [예제5-4] 잘못된 데이터가 입력되었을 때 호출되는 에러 페이지

[WebContent]-[DataError.jsp]

```
<html>
    <head><title>덧셈프로그램 - 에러화면</title></head>
    <body>
        잘못된 데이터가 입력되었습니다.
    </body>
</html>
```

## 5.2 JSP 페이지에서 에러페이지 호출하기(1)

- 익셉션을 발생하는 JSP 페이지에서 `errorPage` 속성을 쓰고 URL 경로를 지정해 놓으면 익셉션 발생시 URL 위치 웹 자원이 호출된다.
- `errorPage`에 `DataError.jsp` 값을 주면 에러가 발생했을 때 같은 디렉토리에 있는 `DataError.jsp`가 호출된다.

### [예제5-5] 두 수를 더하는 JSP 페이지 [page 지시자의 errorPage 애트리뷰트 사용]

[WebContent]-[NewAdder.jsp]

```
<%@ page contentType="text/html; charset=EUC-KR" errorPage="DataError.jsp"%>
<%
    String str1=request.getParameter("num1");
    String str2=request.getParameter("num2");
    int num1=Integer.parseInt(str1);
    int num2=Integer.parseInt(str2);
    int result=num1 + num2;
%>
<html>
    <head><title>덧셈프로그램</title></head>
    <body>
        <%=num1%> + <%=num2%> = <%=result%>
    </body>
</html>
```

## 5.3 JSP 페이지에서 에러페이지 호출하기(2)

- `exception` 내장변수 사용을 위해 `page` 지시자에 `isErrorPage` 속성을 쓰고 `true` 값을 지정한다.
- `exception` 내장변수 `getMessage` 메서드를 호출하여 에러메시지를 가져온다.
- HTTP 상태코드는 HTTP 응답메시지의 시작 행에 표시되는 3자리 숫자로 메시지에 포함된 HTML 문서가 정상적인 실행의 결과인지 에러 발생의 결과인지 구분하는 역할을 한다.
- 웹 브라우저 상태코드 값이 500이면 HTTP 응답메시지에 포함된 HTML 문서를 무시하고 웹 브라우저 자체에 내장된 에러메시지를 출력한다.
- 인위적으로 HTTP 상태 코드의 값을 200으로 바꿔주기 위해서는 `request` 내장 변수에 대해 `setStatus` 메서드를 이용해 변경해 준다.

#### [예제5-6] 잘못된 입력 데이터를 처리하는 에러페이지

[WebContent]-[DataError.jsp]

```
<%@page contentType="text/html; charset=euc-kr" isErrorPage="true" %>
<%response.setStatus(200);%>
<html>
  <head>
    <title>덧셈프로그램 - 에러화면</title>
  </head>

  <body>
    잘못된 데이터가 입력되었습니다.<br><br>
    상세 에러 메시지:<%=exception.getMessage()%>
  </body>
</html>
```

#### [예제5-7] 두 수를 더하는 서블릿 클래스

[Java Resources]-[src]-[NewAdderServlet.java]

```
@WebServlet("/new-adder")
public class NewAdderServlet extends HttpServlet{
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out=response.getWriter();

        String str1=request.getParameter("num1");
        String str2=request.getParameter("num2");

        try{
            int num1=Integer.parseInt(str1);
            int num2=Integer.parseInt(str2);
            int result=num1 + num2;

            out.println("<html>");
            out.println("<head><title>덧셈프로그램</title></head>");

            out.println("<body>");
            out.printf("%d + %d = %d", num1, num2, result);
            out.println("</body>");
            out.println("</html>");

        }catch(NumberFormatException e){
            RequestDispatcher dispatcher=request.getRequestDispatcher("data-error");
            dispatcher.forward(request, response);
        }
    }
}
```

#### [예제5-8] 에러메시지를 출력하는 서블릿 클래스

[Java Resources]-[src]-[DataErrorServlet.java]

```
@WebServlet("/data-error")
public class DataErrorServlet extends HttpServlet{
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        response.setContentType("text/html; charset=UTF-8");

        PrintWriter out=response.getWriter();

        out.println("<html>");
        out.println("<head><title>덧셈프로그램 - 에러화면</title></head>");

        out.println("<body>");
        out.println("잘못된 데이터가 입력되었습니다.");
        out.println("</body>");
        out.println("</html>");
    }
}
```

## 5.4 web.xml 파일에 에러 페이지 등록하기

- web.xml 파일에는 타입별로 에러페이지를 등록해 놓을 수 있다. 해당 타입의 익셉션이 발생하면 자동으로 등록된 에러페이지가 호출된다.

[WebContent]-[WEB-INF]-[web.xml]

```
<web-app>
  <error-page>
    <exception-type>java.lang.NumberFormatException</exception-type>
    <location>/NumberFormatError.jsp</location>
  </error-page>
</web-app>
```

[예제5-9] 두 수의 곱을 구하는 JSP 페이지 [java.lang.NumberFormatException 발생가능]

[WebContent]-[Multiplier.jsp]

```
<%
String str1=request.getParameter("num1");
String str2=request.getParameter("num2");
int num1=Integer.parseInt(str1);
int num2=Integer.parseInt(str2);
int result=num1 * num2;
%>
<html>
  <head><title>곱셈 프로그램</title></head>
  <body>
    <%=num1%> * <%=num2%> = <%=result%>
  </body>
</html>
```

[예제5-10] java.lang.NumberFormatException이 발생했을 때 호출되는 에러페이지

[WebContent]-[NumberFormatError.jsp]

```
<%@page contentType="text/html; charset=euc-kr" isErrorPage="true"%>
<%response.setStatus(200);%>
<html>
  <head><title>숫자 포맷 에러</title></head>
  <body>
    잘못된 데이터가 입력되었습니다.<br><br>
    상세 에러 메시지:<%=exception.getMessage()%>
  </body>
</html>
```

[예제5-11] 두 수의 몫을 구하는 JSP 페이지 [java.lang.NumberFormatException 발생가능]

[WebContent]-[WEB-INF]-[web.xml]

```
<error-page>
  <exception-type>java.lang.ArithmeticException</exception-type>
  <location>/ArithmeticError.jsp</location>
</error-page>
```

[WebContent]-[Divider.jsp]

```
<%
String str1=request.getParameter("num1");
String str2=request.getParameter("num2");
int num1=Integer.parseInt(str1);
int num2=Integer.parseInt(str2);
int result=num1 / num2;
%>
<html>
  <head><title>나눗셈 프로그램</title></head>
  <body>
    <%=num1%> / <%=num2%> = <%=result%>
  </body>
</html>
```

[예제5-12] java.lang.ArithmeticException이 발생했을 때 호출되는 에러페이지

[WebContent]-[ArithmeticError.jsp]

```
<%@page contentType="text/html; charset=euc-kr" isErrorPage="true"%>
<%response.setStatus(200);%>
<html>
  <head><title>산술 연산 에러</title></head>
  <body>
    산술 연산 도중에 에러가 발생했습니다.<br><br>
    상세에러메시지: <%=exception.getMessage()%>
  </body>
</html>
```

[예제5-13] 두 수의 몫을 구하는 서블릿 클래스

[Java Resources]-[src]-[DividerServlet.java]

```
@WebServlet("/divider")
public class DividerServlet extends HttpServlet{
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out=response.getWriter();

        String str1=request.getParameter("num1");
        String str2=request.getParameter("num2");
        int num1=Integer.parseInt(str1);
        int num2=Integer.parseInt(str2);
        int result=num1 / num2;

        out.println("<html>");
        out.println("<head><title>나눗셈프로그램</title></head>");
        out.println("<body>");
        out.printf("%d / %d = %d", num1, num2, result);
        out.println("</body>");
        out.println("</html>");
    }
}
```

## 5.5 Http 상태 코드별로 에러 페이지를 등록하는 방법

web.xml 파일에는 상태코드 별로 에러페이지를 등록할 수 있다. web.xml파일의 루트 엘리먼트 아래에 <error-page>라는 서브엘리먼트를 쓰고 그 안에 다시 <error-code>와 <location>이라는 두 개의 서브엘리먼트를 쓰는 것이다.

[예제5-14] 404 에러를 처리하는 에러 페이지

[WebContent]-[WEB-INF]-[web.xml]

```
<error-page>
  <error-code>404</error-code>
  <location>/NotFoundError.jsp</location>
</error-page>
```

[WebContent]-[NotFoundError.jsp]

```
<%response.setStatus(200);%>
<html>
  <head><title>페이지 없음 에러</title></head>
  <body>
    해당페이지를 찾을 수 없습니다.
  </body>
</html>
```



## 제6장 서블릿의 라이프 사이클

### 6.1 서블릿의 라이프 사이클

- 서블릿 클래스는 웹 브라우저에 의해 바로 호출되는 것이 아니라 서블릿 클래스로부터 서블릿 객체가 만들어지고 그 객체가 웹 컨테이너에 의해 초기화된 다음에 호출된다.
- 웹 컨테이너는 자신이 종료되기 전이나 웹 애플리케이션을 리로드(unload)하기 전에 그에 속하는 모든 서블릿을 제거한다.
- 서블릿 라이프 사이클 전체에 한번만 실행되어야 할 코드는 서블릿 클래스 안에 init 메서드와 destroy라는 메서드를 안에 정의한다.

### 6.2 init 메서드의 작성방법

init 메서드는 파라미터가 없는 메서드로 선언해야 하고 리턴 타입은 void로 지정해야하며 public 메서드로 선언해야 한다.

[예제6-1] 피보나치 수열을 출력하는 서블릿 클래스

[Java Resources]-[src]-[FibonacciServlet.java]

```
@WebServlet(urlPatterns= "/fibonacci", loadOnStartup=1)
public class FibonacciServlet extends HttpServlet{
    private BigInteger arr[];

    @Override
    public void init()throws ServletException {
        arr = new BigInteger[100];
        arr[0]=new BigInteger("1");
        arr[1]=new BigInteger("2");
        for(int cnt=2; cnt<arr.length; cnt++){
            arr[cnt]=arr[cnt-2].add(arr[cnt-1]);
        }
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();

        String str=request.getParameter("num");
        int num=Integer.parseInt(str);
        if(num > 100){
            num=100;
        }
        out.println("<html>");
        out.println("<head><title>피보나치 수열</title></head>");
        out.println("<body>");
        for(int cnt=0; cnt < num; cnt++){
            out.println(arr[cnt] + " ");
        }
        out.println("</body>");
        out.println("</html>");
    }
}
```

### 6.3 destroy 메서드의 작성방법

- destroy 메서드의 작성 방법은 init 메서드와 비슷하다. 파라미터가 없고 리턴 타입은 void여야 하며 public 메서드로 선언해야 한다.
- init 메서드와 달리 throws 절을 쓸 수 없다.

[예제6-2] 이름을 입력받는 HTML 문서

[WebContent]-[YourName.html]

```
<html>
<head><title>이름 입력</title></head>
<body>
<h3>이름을 입력하십시오.</h3>
<form action="greeting">
    이름:<input type="text" name="name">
    <input type="submit" value="확인">
</form>
</body>
</html>
```

### [예제6-3] 인사말을 출력하는 서블릿 클래스

[Java Resource]-[src]-[GreetingServlet.java]

```
@WebServlet("/greeting")
public class GreetingServlet extends HttpServlet{
    private PrintWriter logFile;
    public void init() throws ServletException{
        try{
            logFile = new PrintWriter(new FileWriter("c:/data/log.txt", true));
        }catch(Exception e){ throw new ServletException(e); }
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        String name=request.getParameter("name");
        String greeting="안녕하세요, " + name + "님.";
        if(logFile != null){
            Date now=new Date();
            logFile.printf("%TF %TT - %s %n", now, now, name);
        }
        out.println("<html>");
        out.println("<head><title>인사하기</title></head>");
        out.println("<body>");
        out.println(greeting);
        out.println("</body>");
        out.println("</html>");
    }
    public void destroy(){
        if(logFile != null){ logFile.close(); }
    }
}
```

## 6.4 서블릿의 초기화 파라미터

서블릿의 초기화 파라미터는 서블릿 초기화 작업에 필요한 데이터를 의미하며 값을 가져오려면 `getInitParameter`라는 메서드를 호출하면 된다.

### [예제6-4] 인사말을 출력하는 서블릿 클래스 [서블릿의 초기화파라미터 사용]

[Source]-[src]-[GreetingServlet.java] 수정

```
@WebServlet(urlPatterns="/greeting", initParams=@WebInitParam(name="FILE_NAME", value="c:/data/greeting_log.txt"))
public class GreetingServlet extends HttpServlet {
    private PrintWriter logFile;
    public void init() throws ServletException{
        String fileName = getInitParameter("FILE_NAME");
        try{
            logFile = new PrintWriter(new FileWriter(fileName, true));
        }catch(Exception e){ throw new ServletException(e); }
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        String name=request.getParameter("name");
        String greeting="안녕하세요, " + name + "님.";
        if(logFile != null) {
            Date now=new Date();
            logFile.printf("%TF %TT - %s %n", now, now, name);
        }
        out.println("<html>");
        out.println("<head><title>인사하기</title></head>");
        out.println("<body>");
        out.println(greeting);
        out.println("</body>");
        out.println("</html>");
    }
    public void destroy() {
        if(logFile != null) { logFile.close(); }
    }
}
```

## 6.5 jspInit 메서드와 jspDestroy 메서드의 작성방법

- jspInit 메서드는 파라미터가 없는 메서드를 만들고 리턴타입을 void로 지정하고 public 메서드로 선언하지만 throw 절을 쓸 수 없다.
- jspDestory 메서드는 파라미터가 없어야 하고 리턴타입은 void로 지정하며 public 메서드로 선언해야 한다.

[예제6-5] jspInit, jspDestroy 메서드의 사용 예를 보여주는 JSP 페이지

[WebContent]-[DateTime.jsp]

```
<%@ page import="java.io.*, java.util.*"%>
<%!
    private PrintWriter logFile;
    public void jspInit(){
        String fileName="c:/data/datetime_log.txt";
        try{
            logFile=new PrintWriter(new FileWriter(fileName, true));
        }catch(Exception e){
            System.out.printf("%TT - %s 파일을 열 수 없습니다. %n", new Date(), fileName);
        }
    }
%>
<html>
<body>
<%
    Date now=new Date();
    String date=String.format("현재날짜: %TY년 %Tm월 %Te일", now, now, now);
    String time=String.format("현재시각: %T시 %TM분 %TS초", now, now, now);
    out.println(date + "<br>");
    out.println(time + "<br>");
    if(logFile != null) { logFile.printf("%TF %TT에 호출되었습니다. %n", now, now); }
%>
</body>
</html>
<%!
    public void jspDestroy() {
        if(logFile != null){ logFile.close(); }
    }
%>
```

## 6.6 jsp 페이지의 초기화 파라미터

JSP 페이지는 초기화 파라미터를 기재하기 위해서는 web.xml 파일에 등록해야 한다.

[예제6-6] 초기화 파라미터를 가져오는 JSP 페이지

[WebContent]-[DBName.jsp]

```
<html>
<body>
    <%String dbName=getInitParameter("DB_NAME");%>
    데이터베이스 이름:<%=dbName%><br>
</body>
</html>
```

[WebContent]-[WEB-INF]-[web.xml]

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="3.1">
    <servlet>
        <servlet-name>dbname-jsp</servlet-name>
        <jsp-file>/DBName.jsp</jsp-file>
        <init-param>
            <param-name>DB_NAME</param-name>
            <param-value>malldb</param-value>
        </init-param>
    </servlet>
    <servlet-mapping>
        <servlet-name>dbname-jsp</servlet-name>
        <url-pattern>/dbname</url-pattern>
    </servlet-mapping>
</web-app>
```

## 6.7 서블릿의 환경 정보를 가져오는 방법

- 서블릿 클래스나 JSP 페이지의 환경에 관련된 정보는 javax.servlet 패키지의 ServletContext 인터페이스 타입의 객체를 이용해서 얻는다.
- 서블릿 클래스에서 이 타입의 객체를 구하기 위해서는 getServletContext라는 메서드를 호출하면 된다.
- ServletContext객체에 getServerInfo라는 메서드를 호출하면 서블릿이 속하는 웹 서버 종류가 리턴 된다.
- ServletContext객체에 getMajorVersion과 getMinorVersion 메서드를 호출시 서블릿 규격서의 메이저버전과 마이너버전이 리턴 된다.

### [예제6-7] 웹 서버의 정보를 출력하는 JSP 페이지

[Java Resources]-[src]-[ServerInfoServlet.java]

```
@WebServlet("/server-info")
public class ServerInfoServlet extends HttpServlet{
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        ServletContext context = getServletContext();

        String serverInfo = context.getServerInfo();

        int majorVersion = context.getMajorVersion();
        int minorVersion = context.getMinorVersion();

        out.println("<html>");
        out.println("<head><title>웹 서버의 정보</title></head>");

        out.println("<body>");
        out.printf("웹 서버의 종류: %s <br>", serverInfo);
        out.printf("지원하는 서블릿 버전: %d.%d <br>", majorVersion, minorVersion);
        out.println("</body>");
        out.println("</html>");
    }
}
```

### [예제6-8] application 내장 변수의 사용 예를 보여주는 JSP 페이지

[Java Resource]-[src]-[ServerInfo.jsp]

```
<html>
  <head>
    <title>웹 서버의 정보</title>
  </head>
  <body>
    웹서버의 종류: <%=application.getServerInfo()%><BR>
    지원하는 서블릿 버전: <%=application.getMajorVersion()%>.<%=application.getMinorVersion()%><br>
  </body>
</html>
```

### [예제6-9] 웹 애플리케이션의 초기화 파라미터를 읽는 JSP 페이지

[WebContent]-[AppParamTest.jsp]

```
<html>
  <head>
    <title>웹 애플리케이션의 초기화 파라미터</title>
  </head>
  <body>
    데이터베이스이름: <%=application.getInitParameter("DB_NAME")%><br>
  </body>
</html>
```

[WebContent]-[WEB-INF]-[web.xml] 추가

```
<context-param>
  <param-name>DB_NAME</param-name>
  <param-value>bankdb</param-value>
</context-param>
```

## 6.8 로그 메시지를 기록하는 log메서드

- ServletContext 인터페이스의 log 메서드를 이용하면 로그 파일에 메시지를 기록할 수 있다.
- log 메서드는 파라미터로 넘겨준 메시지를 톰캣의 설치 디렉터리 아래의 logs라는 이름의 서브디렉터리 안에 기록한다.

[예제6-10] log 메서드의 사용예를 보여주는 JSP 페이지

[WebContent]-[Hello.jsp]

```
<html>
  <head><title>인사하기</title></head>
  <body>
    안녕하세요, <%=request.getParameter("NAME")%>님
    <%application.log("[인사하기] JSP 페이지가 호출되었습니다.");%>
  </body>
</html>
```

## 6.9 같은 웹 애플리케이션에 속하는 웹 컴포넌트들끼리 데이터를 주고받는 방법

- setAttribute: 이 메서드는 웹 애플리케이션에 할당된 공유 데이터 영역에 데이터를 저장하는 기능을 한다.
- getAttribute: 이 메서드는 그 영역에 있는 데이터를 읽어오는 기능을 한다.
- removeAttribute: 이 메서드는 그 영역의 데이터를 삭제하는 기능을 한다.

[예제6-11] 웹 애플리케이션의 데이터 영역에 데이터를 저장하는 JSP 페이지

[WebContent]-[StoreName.jsp]

```
<%
  String name = request.getParameter("name");
  application.setAttribute("name", name);
%>
<html>
  <head><title>웹 애플리케이션 데이터 저장하기</title></head>
  <body>
    NAME 데이터가 저장되었습니다.
  </body>
</html>
```

[예제6-12] 웹 애플리케이션의 데이터 영역으로부터 데이터를 읽어오는 JSP 페이지

[WebContent]-[ReadName.jsp]

```
<html>
  <head><title>웹 애플리케이션 데이터 조회하기</title></head>
  <body>
    이름: <%=application.getAttribute("name")%>
  </body>
</html>
```

[예제6-13] 웹 애플리케이션의 데이터 영역에 있는 데이터를 삭제하는 JSP 페이지

[WebContent]-[DeleteName.jsp]

```
<%application.removeAttribute("name");%>
<html>
  <head><title>웹 애플리케이션 데이터 삭제하기</title></head>
  <body>
    NAME 데이터가 삭제되었습니다.
  </body>
</html>
```

## 제7장 익스프레션 언어

### 7.1 익스프레션 언어란?

- 익스프레션 언어(expression language)란 식(expression)을 중심으로 코드를 기술하는 언어이다.
- 연산자와 피연산자의 조합을 \${와 }로 둘러싸서 표현한다.

### 7.2 데이터 이름 하나로만 구성된 EL식

데이터 이름 하나로만 구성된 EL식은 가장 간단한 형태의 EL식으로 EL식 안에 기술되는 데이터 이름은 애플릿 이름으로 해석된다.

[예제7-1] 1부터 100까지의 합을 구하는 JSP 페이지

[WebContent]-[HundredResult.jsp]

```
<html>
  <head><title>1부터 100까지의 합</title></head>
  <body>
    1부터 100까지 더한 결과는? ${result}
  </body>
</html>
```

[WebContent]-[Hundred.jsp]

```
<%
  int sum = 0;
  for(int cnt = 1; cnt <= 100; cnt++) {
    sum += cnt;
  }
  request.setAttribute("result", sum);
  RequestDispatcher dispatcher = request.getRequestDispatcher("HundredResult.jsp");
  dispatcher.forward(request, response);
%>
```

[예제7-2] 1부터 1000까지의 합을 구하는 JSP 페이지

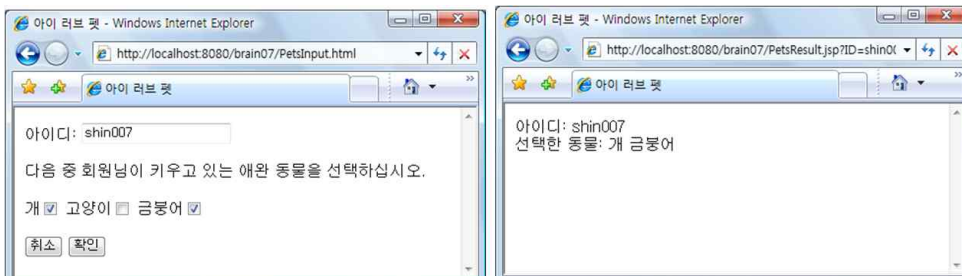
[WebContent]-[Thousand.jsp]

```
<%
  int sum = 0;
  for(int cnt = 1; cnt <= 1000; cnt++){
    sum += cnt;
  }
  pageContext.setAttribute("result", sum);
%>
<html>
  <head><title>1부터 1000까지의 합</title></head>
  <body>
    1부터 1000까지 더한 결과는? ${result}
  </body>
</html>
```

### 7.3 익스프레션 언어의 내장 객체

- param은 웹 브라우저에서 <form> 엘리먼트를 통해 입력된 데이터를 가져올 때 사용하는 내장 객체이다.
- <form> 엘리먼트를 통해 똑같은 이름의 데이터가 여러개 입력되는 경우도 있는데 그럴 때는 paramValues 내장 객체를 사용하면 된다.

[예제7-3] 데이터를 입력받는 HTML 문서



[WebContent]-[PetsInput.html]

```
<head><title>아이 러브 펫</title></head>
<body>
  <form action="PetsResult.jsp">
    아이디: <input type="text" name="id"><br><br>
    다음 중 회원님이 키우고 있는 애완동물을 선택하십시오.<br><br>
    개<input type="checkbox" name="animal" value="개">
    고양이<input type="checkbox" name="animal" value="고양이">
    금붕어<input type="checkbox" name="animal" value="금붕어"><br><br>
    <input type="reset" value="취소">
    <input type="submit" value="확인">
  </form>
</body>
```

[예제7-4] 입력된 데이터를 출력하는 JSP 페이지

[WebContent]-[PetsResult.jsp]

```
<html>
  <head><title>아이 러브 펫</title></head>
  <body>
    아이디:${param.id}<br>
    선택한동물:
    ${paramValues.animal[0]}
    ${paramValues.animal[1]}
    ${paramValues.animal[2]}
  </body>
</html>
```

[예제7-5] 쿠키 데이터를 가져다가 출력하는 EL 식의 예

[WebContent]-[CookieDataWriter.jsp]

```
<%
  Cookie cookie = new Cookie("name", "John");
  response.addCookie(cookie);
%>
<html>
  <head><title>쿠키 데이터 저장 프로그램</title></head>
  <body>쿠키값이 설정되었습니다.</body>
</html>
```

[WebContent]-[CookieDataReader.jsp]

```
<html>
  <head><title>쿠키 데이터 출력 프로그램</title></head>
  <body>name 쿠키 데이터의 값은? ${cookie.name.value}</body>
</html>
```

[예제7-6] 웹 애플리케이션의 초기화 파라미터 값을 출력하는 JSP 페이지

[WebContent]-[InitParam.jsp]

```
<html>
  <head><title>애플리케이션 초기화 파라미터 예제</title></head>
  <body>DB_NAME 초기화 파라미터의 값은? ${initParam.DB_NAME}</body>
</html>
```

[WebContent]-[WEB-INF]-[web.xml]

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="3.1">
  <context-param>
    <param-name>DB_NAME</param-name>
    <param-value>bankdb</param-value>
  </context-param>
</web-app>
```

### [예제7-7] pageContext 내장 객체의 사용 예

[WebContent]-[PageContext.jsp]

```
<html>
  <body>
    요청URL: ${pageContext.request.requestURI}
  </body>
</html>
```

## 7.3 산술 연산자, 비교 연산자, 논리 연산자, 조건 연산자

특정 기호들은 잘못된 해석 방지를 위해 익스프레션 언어에서는 기호로 된 연산자들과 똑같은 기능을 하는 영어단어 연산자들이 존재한다.

### [예제7-8] 익스프레션 언어의 연산자 사용 예(1)

[WebContent]-[operators.jsp]

```
<html>
  <body>
    x = ${param.num1}, y = ${param.num2}<br><br>
    x + y = ${param.num1 + param.num2}<br>
    x - y = ${param.num1 - param.num2}<br>
    x * y = ${param.num1 * param.num2}<br>
    x / y = ${param.num1 / param.num2}<br>
    x % y = ${param.num1 % param.num2}<br><br>

    x가 더 큼니까? ${param.num1 - param.num2 > 0}<br><br>
    y가 더 큼니까? ${param.num2 - param.num2 < 0}<br><br>
    x와 y가 모두 양수입니까? ${param.num1 > 0} && (param.num2 > 0)<br><br>
    x와 y가 같습니까? ${param.num1 == param.num2 ? "예" : "아니오"}<br><br>
  </body>
</html>
```

### [예제7-9] 익스프레션 언어의 연산자 사용 예(2)

[WebContent]-[CharOperators.jsp]

```
<html>
  <body>
    ${param.num1}을 ${param.num2}로 나눈 몫은? ${param.num1 div param.num2}<br>
    나머지는? ${param.num1 mod param.num2}<br><br>
    둘 다 양수입니까? ${param.num1 > 0} and (param.num2 > 0)<br>
    둘 다 음수입니까? ${param.num1 < 0} and (param.num2 < 0)<br>
  </body>
</html>
```

### [예제7-10] 익스프레션 언어의 연산자 사용 예(3)

[WebContent]-[StringOperators.jsp]

```
<html>
  <body>
    입력 문자열: ${param.str1}, ${param.str2}<br><br>
    두 문자열이 같습니까? ${param.str1 == param.str2}<br>
    어느 문자열이 먼저입니까? ${param.str1 < param.str2 ? param.str1 : param.str2}
  </body>
</html>
```

### [예제7-11] 앵퍼티 연산자의 사용 예

[WebContent]-[EmptyOperator.jsp]

```
<html>
  <body>
    안녕하세요, ${empty param.ID ? "guest" : param.ID}님
  </body>
</html>
```



## 7.4 대괄호 연산자와 마침표 연산자

- 배열과 java.util.List 객체의 데이터 항목을 가리킬 때는 반드시 대괄호 연산자를 이용해야 한다.
- java.util.Map 객체의 데이터 항목과 자바빈 프로퍼티를 가리킬 때는 대괄호 연산자와 마침표 연산자 중 하나를 사용할 수 있다.

### [예제7-12] 배열 항목을 출력하는 EL 식의 예

[WebContent]-[Winners.jsp]

```
<%
    String winners[] = new String[3];
    winners[0] = "이수현"
    winners[1] = "정세훈"
    winners[2] = "김진희"
    request.setAttribute("winners", winners);
    RequestDispatcher dispatcher = request.getRequestDispatcher("WinnersView.jsp");
    dispatcher.forward(request, response);
%>
```

[WebContent]-[WinnersView.jsp]

```
<html>
    <head><title>우승자 명단</title></head>
    <body>
        <h3>우승자 명단</h3>
        1등. ${winners[0]}<br>
        2등. ${winners[1]}<br>
        3등. ${winners[2]}<br>
    </body>
</html>
```

### [예제7-13] List 객체의 항목을 출력하는 EL 식의 예

[WebContent]-[Fruits.jsp]

```
<%@page import="java.util.*"%>
<%
    ArrayList<String> items = new ArrayList<String>();
    items.add("딸기");
    items.add("오렌지");
    items.add("복숭아");
    request.setAttribute("fruits", items);
    RequestDispatcher dispatcher = request.getRequestDispatcher("FruitsView.jsp");
    dispatcher.forward(request, response);
%>
```

[WebContent]-[FruitsView.jsp]

```
<html>
    <head><title>인기 상품 목록</title></head>
    <body>
        <h3>이달에 가장 많이 팔린 과일입니다.</h3>
        1위. ${fruits[0]}<br>
        2위. ${fruits[1]}<br>
        3위. ${fruits[2]}<br>
    </body>
</html>
```

### [예제7-14] Map 객체의 항목을 출력하는 EL 식의 예

[WebContent]-[AddressView.jsp]

```
<html>
    <head><title>주소록</title></head>
    <body>
        ${param.name}의 주소는? ${address[param.name]}
    </body>
</html>
```

[예제7-15] 자바빈 클래스의 예

[Java Resources]-[src]-[mall]-[ProductInfo.java]

```
package mall;
public class ProductInfo{
    private String name;
    private int value;
    public ProductInfo(){
    }
    public void setName(String name){
        this.name = name;
    }
    public String getName(){
        return name;
    }
    public void setPrice(int price){
        value = price;
    }
    public int getPrice(){
        return value;
    }
}
```

---

[예제7-16] 자바빈의 프로퍼티를 출력하는EL 식의 예

[WebContent]-[ProductInfo.jsp]

```
<%@page import="mall.ProductInfo"%>
<%
    ProductInfo product = new ProductInfo();
    product.setName("초코케이크 3호");
    product.setPrice(20000);
    request.setAttribute("product", product);
    RequestDispatcher dispatcher = request.getRequestDispatcher("ProductInfoView.jsp");
    dispatcher.forward(request, response);
%>
```

---

[WebContent]-[ProductInfoView.jsp]

```
<html>
<head><title>상품 정보</title></head>
<body>
    <h3>상품 정보</h3>
    상품명: ${product.name}<br>
    가격: ${product.price}원<br>
</body>
</html>
```

---

## 제8장 표준 액션

### 8.1 <jsp:include> 표준 액션의 사용 방법

- <jsp:include>는 JSP 페이지에서 다른 웹 자원(JSP 페이지, HTML 문서 등)을 포함시키고자 할 때 사용하는 표준 액션이다.
- 액션 태그는 XML 문법을 따르므로 단독 태그일 경우에는 위와 같이 '/'로 끝나도록 만들어야 한다.

[예제8-1] <jsp:include> 액션의 사용 예를 보여주는 예제(1)

[WebContent]-[BooksInfo.jsp]

```
<html>
  <head><title>책 소개</title></head>
  <body>
    <h3>책 소개</h3>
    제목: 뇌를 자극하는 Java 프로그래밍<br>
    저자: 김운명<br>
    페이지수: 908<br><br>
    <jsp:include page= "Copyright.html"/>
  </body>
</html>
```

[WebContent]-[Copyright.html]

```
<font size=2>CopyRight 이 웹 페이지의 저작권은 한빛미디어(주)에 있습니다.</font>
```

[예제8-2] <jsp:include> 액션의 사용 예를 보여주는 예제(2)

[WebContent]-[Winner.jsp]

```
<html>
  <head><title>당첨자 명단</title></head>
  <body>
    <h3>당첨자 명단</h3>
    14553 연홍부<br>
    63563 심청<br>
    73992 이몽룡<br><br>
    <jsp:include page="Now.jsp"/>
  </body>
</html>
```

[WebContent]-[Now.jsp]

```
<%@page import="java.util.*"%>
<%
  GregorianCalendar now = new GregorianCalendar();
  String date = String.format("%TY년 %Tm월 %Td일", now, now, now);
  String time = String.format("%Tp %TR", now, now);
%>
[현재 시각] <%=date%> <%=time%>
```

### 8.2 <jsp:forward> 표준 액션의 사용 방법

<jsp:forward>는 JSP 페이지에서 다른 JSP 페이지로 제어를 넘기고자 할 때 사용하는 표준 액션이다.

[예제8-3] <jsp:forward> 액션의 사용 예를 보여주는 예제

[WebContent]-[Hundred.jsp]

```
<%
  int sum = 0;
  for (int cnt = 1; cnt <= 100; cnt++){
    sum += cnt;
  }
  request.setAttribute("result", sum);
%>
<jsp:forward page="HundredResult.jsp"/>
```

[WebContent]-[HundredResult.jsp]

```
<html>
  <head><title>1부터 100까지의 합</title></head>
  <body>
    1부터 100까지 더한 결과는? ${result}
  </body>
</html>
```

### 8.3 자바빈 관련 표준 액션의 기초 사용 방법

- <jsp:useBean> 표준 액션을 이용하면 자바 코드를 작성하지 않고도 자바빈 객체를 만들 수 있다.
- <jsp:setProperty> 표준 액션을 이용하면 set- 메서드를 직접 호출하지 않고도 프로퍼티 값을 설정할 수 있다.
- <jsp:getProperty> 표준 액션을 이용하면 get- 메서드를 직접 호출하지 않고도 프로퍼티 값을 가져올 수 있다.

[예제8-4] 개인 정보를 표현하는 자바빈 클래스

[Java Resources]-[src]-[mall]-[PersonalInfo.java]

```
package mall;
public class PersonalInfo{
    private String name;    //이름
    private char gender;    //성별
    private int age;        //나이
    setter()/getter()/toString() 생략
}
```

[예제8-5] 표준 액션을 이용해서 자바빈 프로퍼티를 설정하고 가져오는 JSP 페이지(1)

[WebContent]-[CustomerInfo.jsp]

```
<html>
  <head><title>회원 정보</title></head>
  <body>
    <jsp:useBean class="mall.PersonalInfo" id="pinfo"/>
    <jsp:setProperty name="pinfo" property="name" value="김연희"/>
    <jsp:setProperty name="pinfo" property="gender" value="여"/>
    <jsp:setProperty name="pinfo" property="age" value="29"/>
    이름: <jsp:getProperty name="pinfo" property="name"/><br>
    성별: <jsp:getProperty name="pinfo" property="gender"/><br>
    나이: <jsp:getProperty name="pinfo" property="age"/>
  </body>
</html>
```

[예제8-6] 표준 액션을 이용해서 자바빈 프로퍼티를 설정하고 가져오는 JSP 페이지(2)

[WebContent]-[CustomerInfoSaver.jsp]

```
<jsp:useBean class="mall.PersonalInfo" id="pinfo" scope="request"/>
<jsp:setProperty name="pinfo" property="name" value="김현수"/>
<jsp:setProperty name="pinfo" property="gender" value="남"/>
<jsp:setProperty name="pinfo" property="age" value="23"/>
<jsp:forward page="CustomerInfoViewer.jsp"/>
```

[WebContent]-[CustomerInfoViewer.jsp]

```
<html>
  <head><title>회원 정보</title></head>
  <body>
    <jsp:useBean class="mall.PersonalInfo" id="pinfo" scope="request"/>

    이름: <jsp:getProperty name="pinfo" property="name"/><br>
    성별: <jsp:getProperty name="pinfo" property="gender"/><br>
    나이: <jsp:getProperty name="pinfo" property="age"/>
  </body>
</html>
```

[예제8-7] 서블릿 클래스가 JSP 페이지로 자바빈 객체를 넘겨주는 예

[Java Resource]-[src]-[mall]-[PersonalInfoServlet.java]

```
package mall;
@WebServlet("/pinfo")
public class PersonalInfoServlet extends HttpServlet{
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException{
        PersonalInfo obj = new PersonalInfo();
        obj.setName("이정호");
        obj.setGender('남');
        obj.setAge(24);
        request.setAttribute("pinfo", obj);
        RequestDispatcher dispatcher = request.getRequestDispatcher("CustomerInfoViewer.jsp");
        dispatcher.forward(request, response);
    }
}
```

## 8.4 웹 브라우저로부터 입력된 데이터를 자바빈 프로퍼티로 설정하는 방법

- 익스프레션 언어의 param 내장 객체를 사용하면 입력 데이터를 바로 가져옴과 동시에 타입 변환도 자동으로 이루어져서 편리하다.
- <jsp:setProperty> 액션에 value 속성 대신 param 속성을 사용하면 입력 데이터의 이름을 직접 지정할 수 있다.

[예제8-8] 웹 브라우저로부터 입력된 데이터를 자바빈 프로퍼티 값으로 설정하는 예(1)

[WebContent]-[NewCustomerInfo.java]

```
<html>
<body>
    <jsp:useBean class="mall.PersonalInfo" id="pinfo"/>
    <jsp:setProperty name="pinfo" property="name" param="name"/>
    <jsp:setProperty name="pinfo" property="gender" param="gender"/>
    <jsp:setProperty name="pinfo" property="age" param="age"/>
    이름: <jsp:getProperty name="pinfo" property="name"/><br>
    성별: <jsp:getProperty name="pinfo" property="gender"/><br>
    나이: <jsp:getProperty name="pinfo" property="age"/>
</body>
</html>
```

[예제8-9] 웹 브라우저로부터 입력된 데이터를 자바빈 프로퍼티 값으로 설정하는 예(2)

[WebContent]-[NewerCustomerInfo.jsp]

```
<html>
<body>
    <jsp:useBean class="mall.PersonalInfo" id="pinfo"/>
    <jsp:setProperty name="pinfo" property="*/>
    이름: <jsp:getProperty name="pinfo" property="name"/><br>
    성별: <jsp:getProperty name="pinfo" property="gender"/><br>
    나이: <jsp:getProperty name="pinfo" property="age"/>
</body>
</html>
```

## 8.5 자바빈의 다형성을 활용하는 방법

클래스들이 서로 상속 관계를 맺고 있으면 서브클래스 타입의 객체를 슈퍼클래스 타입의 변수에 대입해서 사용할 수 있다.

[예제8-10] 상속 관계를 갖는 3개의 자바빈 클래스

[Java Resources]-[src]-[mall]-[ProductInfo.class]

```
package mall;
public class ProductInfo{
    private String code; //제품코드
    private String name; //제품명
    private int price; //가격
    setter()/getter()/toString() 생략
}
```

[Java Resources]-[src]-[mall]-[BookInfo.java]

```
package mall;

public class BookInfo extends ProductInfo{
    private short page;    //페이지수
    private String writer; //저자
    setter()/getter()/toString() 생략
}
```

[Java Resources]-[src]-[mall]-[clothingInfo.java]

```
package mall;

public class ClothingInfo extends ProductInfo{
    private char size;    //사이즈(L/M/S)
    private String color; //색상
    setter()/getter()/toString() 생략
}
```

[예제8-11] 표준 액션을 이용해서 자바빈 클래스의 다형성을 활용 하는 예

[WebContent]-[BookInfoSaver.jsp]

```
<jsp:useBean class="mall.BookInfo" id="pinfo" scope="request"/>
<jsp:setProperty name="pinfo" property="code" value="50001"/>
<jsp:setProperty name="pinfo" property="name" value="의뢰인"/>
<jsp:setProperty name="pinfo" property="price" value="9000"/>
<jsp:setProperty name="pinfo" property="writer" value="존 그리섬"/>
<jsp:setProperty name="pinfo" property="page" value="704"/>

<html>
    <head><title>책 정보 관리</title></head>
    <body>
        책정보가저장되었습니다. <br>
        -----<br>
        <h3>제품 개략 정보</h3>
        <jsp:include page="ProductInfo.jsp"/>
    </body>
</html>
```

[WebContent]-[ProductInfo.jsp]

```
<jsp:useBean class="mall.ProductInfo" id="pinfo" scope="request"/>
코드: <jsp:getProperty name="pinfo" property="code"/><BR>
제품명:<jsp:getProperty name="pinfo" property="name"/><BR>
가격: <jsp:getProperty name="pinfo" property="price"/><BR>
```

[WebContent]-[ClothingInfoSaver.jsp]

```
<jsp:useBean class="mall.ClothingInfo" id="pinfo" scope="request"/>
<jsp:setProperty name="pinfo" property="code" value="70002"/>
<jsp:setProperty name="pinfo" property="name" value="반팔 티셔츠"/>
<jsp:setProperty name="pinfo" property="price" value="15000"/>
<jsp:setProperty name="pinfo" property="size" value="M"/>
<jsp:setProperty name="pinfo" property="color" value="베이지"/>

<html>
    <head><title>의류 정보 관리</title></head>
    <body>
        의류정보가저장되었습니다.<br>
        -----<br>
        <h3>제품 개략 정보</H3>
        <jsp:include page="ProductInfo.jsp"/>
    </body>
</html>
```

## 제9장 JSTL 사용하기

### 9.1 JSTL(JSP Standard Tag Library)

- JSTL을 이용해 간단한 프로그램 로직의 구사, 다른 JSP페이지 호출, 날짜/시간/숫자의 포맷, 데이터베이스로의 입력/수정/삭제/조회, XML 문서의 처리, 문자열을 처리하는 함수 호출 기능들이 가능하다.
- JSTL을 설치하려면 JSTL라이브러리 파일을 다운로드 받은 후 특정 폴더에 아래와 같이 복사해 둔다.
  - 1) <http://jakarta.apache.org/>에서 jstl.jar와 standard.jar 파일을 다운로드한다.
  - 2) [webContent]-[WEB-INF]-[lib] 폴더에 위 두 개의 파일을 복사해 둔다.

#### [예제9-1] <c:set> 커스텀 액션의 사용 예

[WebContent]-[Multiply.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>

<c:set var="num1" value="7"/>
<c:set var="num2" value="9"/>
<c:set var="result" value="${num1*num2}"/>

<html>
  <head><title>곱셈 프로그램</title></head>
  <body>
    ${num1}과 ${num2}의 곱은? ${result}
  </body>
</html>
```

#### [예제9-2] <c:set> 커스텀 액션의 사용 예

[WebContent]-[ProductInfo.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<c:set var="code" value="80012" scope="request"/>
<c:set var="name" value="온습도계" scope="request"/>
<c:set var="price" value="15000" scope="request"/>
<jsp:forward page="ProductInfoView.jsp"/>
```

[WebContent]-[ProductInfoView.jsp]

```
<html>
  <head><title>상품 정보</title></head>
  <body>
    <h3>상품정보</h3>
    상품코드: ${code}<br>
    상품명: ${name}<br>
    단가: ${price}원<br>
  </body>
</html>
```

#### [예제9-3] <c:if> 커스텀 액션의 사용 예

[WebContent]-[Maximum.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
  <head><title>최대값 구하기</title></head>
  <body>
    최대값:
    <c:if test="${param.num1 - param.num2 >= 0}">
      ${param.num1}
    </c:if>
    <c:if test="${param.num1 - param.num2 < 0}">
      ${param.num2}
    </c:if>
  </body>
</html>
```

[예제9-4] <c:choose> 커스텀 액션의 사용 예

- <c:when>, <c:otherwise> 커스텀 액션과 함께 사용되며, 두 커스텀 액션은 <c:otherwise>는 아무 조건도 만족하지 않을 때 실행된다.
- <c:choose> 커스텀 액션은 switch문과 비슷하지만 변수의 이름이 아니라 조건식을 <c:when>의 test 속성에 EL식 형태로 지정한다.

[WebContent]-[Greeting.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
  <head><title>인사하기</title></head>
  <body>
    <c:choose>
      <c:when test="${param.num == 0}">
        처음봬습니다.<br>
      </c:when>
      <c:when test="${param.num == 1}">
        반갑습니다.<br>
      </c:when>
      <c:otherwise>
        안녕하세요.<br>
      </c:otherwise>
    </c:choose>
  </body>
</html>
```

[예제9-5] <c:forEach> 커스텀 액션의 사용 예(1)

- <c:forEach>은 자바의 for 문에 해당하며 begin과 end라는 이름의 속성을 쓰고 거기에 각각 카운터 변수의 시작과 끝 값을 지정한다.
- 반복 출력할 코드 안에서 카운터 변수의 값을 사용해야 할 경우에는 <c:forEach> 태그 안에 var라는 속성을 쓰고 변수이름을 지정한다.
- 카운터 변수의 값은 기본적으로 1씩 증가하지만 그 값을 바꾸려면 <c:forEach> 태그에 step이라는 속성을 추가하고 증가치를 지정한다.

[WebContent]-[Echo.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
  <head><title>메아리</title></head>
  <body>
    <c:forEach var="cnt" begin="1" end="5">
      <font size=${cnt}> 야~호~ </font><br>
    </c:forEach>
  </body>
</html>
```

[예제9-6] <c:forEach> 커스텀 액션의 사용 예(2)

- <c:forEach> 커스텀 액션의 items 속성을 이용하면 여러 개의 항목으로 구성된 데이터를 순서대로 출력하는 일도 할 수 있다.
- <c:forEach var="str" items="\${arr}"> var은 각 항목을 저장할 변수이고 arr은 배열의 이름이다.

[WebContent]-[LunchMenu.jsp]

```
<%
  String arr[] = {"불고기 백반", "오프라이스", "콩국수"};
  request.setAttribute("menu", arr);
%>
<jsp:forward page="LunchMenuView.jsp"/>
```

[WebContent]-[LunchMenuView.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
  <head><title>구내 식당</title></head>
  <body>
    <h3>오늘의 점심 메뉴입니다.</h3>
    <ul>
      <c:forEach var="dish" items="${menu}">
        <li>${dish}</li>
      </c:forEach>
    </ul>
  </body>
</html>
```



[예제9-7] <c:catch> 커스텀 액션의 사용 예

[WebContent]-[Divide.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%
    String str1 = request.getParameter("num1");
    String str2 = request.getParameter("num2");
    int num1 = Integer.parseInt(str1);
    int num2 = Integer.parseInt(str2);
%>
<html>
<head><title>나눗셈 프로그램</title></head>
<body>
    <c:catch var="e">
        <% int result = num1 / num2; %>
        나눗셈의 결과는? <%= result%>
    </c:catch>

    <c:if test="${e != null}">에러 메시지: ${e.message}</c:if>
</body>
</html>
```

[예제9-8] <c:redirect> 커스텀 액션의 사용 예

[WebContent]-[Redirect.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<c:redirect url="Multiply.jsp">
    <c:param name="num1" value="5"/>
    <c:param name="num2" value="25"/>
</c:redirect>
```

[예제9-9] <c:url> 커스텀 액션의 사용 예

[WebContent]-[NewRedirect.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<c:url var="next" value="Divide.jsp">
    <c:param name="num1" value="100"/>
    <c:param name="num2" value="25"/>
</c:url>
<c:redirect url="${next}"/>
```

[예제9-10] <c:out> 커스텀 액션의 사용 예(1)

[WebContent]-[HtmlUsage.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<head><title>HTML 문법 설명</title></head>
<body>
    <h3>FONT 태그에 대하여</h3>
    <c:out value="<font size=7>커다란 글씨</font>는 다음과 같은 출력을 합니다."/><br><br>
    <c:out value="<font size=7>커다란 글씨</font>" escapeXml="false"/>
</body>
</html>
```

[예제9-11] <c:out> 커스텀 액션의 사용 예(2)

[WebContent]-[Hello.jsp]

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<head><title>간단한 인사</title></head>
<body>
    안녕하세요, <c:out value="${param.ID}" default="guest"/>님
</body>
</html>
```

[예제9-12] <fmt:formatDate> 커스텀 액션의 사용 예(1)

[WebContent]-[Now.jsp]

```
<%@page import="java.util.*"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="date" value="%=new Date()%" />
<html>
  <head><title>현재의 시각</title></head>
  <body>
    [오늘의 날짜] <fmt:formatDate value="${date}" /><br>
    [현재의 시각] <fmt:formatDate value="${date}" type="time" />
  </body>
</html>
```

[예제9-13] <fmt:formatDate> 커스텀 액션의 사용 예(2)

[WebContent]-[DateTimeFormat.jsp]

```
<%@page import="java.util.*"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="date" value="%=new Date()%" />
<html>
  <head><title>현재의 시각</title></head>
  <body>
    [S]<fmt:formatDate value="${date}" type="both" dateStyle="short" timeStyle="short" /><br>
    [M]<fmt:formatDate value="${date}" type="both" dateStyle="medium" timeStyle="medium" /><br>
    [L]<fmt:formatDate value="${date}" type="both" dateStyle="long" timeStyle="long" /><br>
    [F]<fmt:formatDate value="${date}" type="both" dateStyle="full" timeStyle="full" />
  </body>
</html>
```

[예제9-14] <fmt:formatDate> 커스텀 액션의 사용 예(3)

[WebContent]-[DateTimePattern.jsp]

```
<%@page import="java.util.*"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<c:set var="date" value="%=new Date()%" />
<html>
  <head><title>현재의 시각</title></head>
  <body>
    [오늘의 날짜]<fmt:formatDate value="${date}" type="date" pattern="yyyy/MM/dd (E)" /><br>
    [현재의 시각]<fmt:formatDate value="${date}" type="time" pattern="(a) hh:mm:ss" />
  </body>
</html>
```

[예제9-15] <fmt:formatNumber> 커스텀 액션의 사용 예(1)

[WebContent]-[NumberFormat.jsp]

```
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
  <head><title>숫자 포맷</title></head>
  <body>
    첫 번째 수:<fmt:formatNumber value="1234500" groupingUsed="true" /><br>
    두 번째 수:<fmt:formatNumber value="3.14158" pattern="###.###" /><br>
    세 번째 수:<fmt:formatNumber value="10.5" pattern="#.00" />
    네 번째 수:<fmt:formatNumber value="1234500" pattern="###,###" />
  </body>
</html>
```

[예제9-16] <fmt:formatNumber> 커스텀 액션의 사용 예(2)

[WebContent]-[NumberType.jsp]

```
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
  <head>
    <title>숫자 포맷</title>
  </head>
  <body>
    금액: <fmt:formatNumber value="1000000" type="currency" currencySymbol="₩"/>
    <br>
    퍼센트: <fmt:formatNumber value="0.99" type="percent"/>
  </body>
</html>
```

## 9.2 JTL 함수 라이브러리 사용하기

함수	제공하는 기능
substring(str, index1, index2)	str의 index1부터 index2 -1까지의 부분 문자열을 리턴
toUpperCase(str)	모든 소문자를 대문자로 치환한 값을 리턴
toLowerCase(str)	모든 대문자를 소문자로 치환한 값을 리턴
trim(str)	문자열에서 앞뒤 공백 문자를 제거한 결과를 리턴
replace(str, src, dest)	str 문자열에 포함된 src를 모두 dest로 치환한 결과를 리턴
indexOf(str1, str2)	str1에 포함된 str2의 시작 인덱스를 리턴
split(str1, str2)	str1을 str2를 기준으로 분리해서 만든 부분 문자열들의 배열을 리턴
join(arr, str2)	arr배열의 모든 항목을 합쳐서 리턴, 항목 사이에는 str2가 들어감
length(obj)	obj가 문자열이면 문자열의 길이, List나 Collection이면 항목의 수를 리턴

[예제9-17] 함수 라이브러리의 사용 예

[WebContent]-[VariousGreetings.jsp]

```
<%@page import="java.util.*"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<c:set var="greeting" value="How Are You?"/>
<html>
  <head>
    <title>여러 가지 인사말</title>
  </head>
  <body>
    본래의 문자열 출력: ${greeting}
    <br>
    모두 대문자로 변경하여 출력: ${fn:toUpperCase(greeting)}
    <br>
    모두 소문자로 변경하여 출력: ${fn:toLowerCase(greeting)}
    <br>
    Are의 위치는 몇 번째에 위치해있는가 ? ${fn:indexOf(greeting, "Are")}
    <br>
    Are를 Were로 바꾸어 출력: ${fn:replace(greeting, "Are ", "Were")}
    <br>
    문자열의 길이는? ${fn:length(greeting)} <br>
  </body>
</html>
```

## 제10장 데이터베이스 사용하기

### 10.1 데이터베이스 만들기

- MySQL 계정 생성: Workbench 실행 후 'root' 계정으로 로그인 후 새로운 계정을 생성한다.

```
create user 'userName'@'localhost' identified by 'password';
create user 'web'@'localhost' identified by 'pass';
```

- 사용자 확인

```
use mysql;
select user, host from user;
```

- MySQL 새로운 스키마(데이터베이스) 생성

```
create database databaseName;
create database webdb;
```

- 사용자에게 데이터베이스 권한 부여 및 데이터베이스 Open

```
grant all privileges on databaseName.* to 'userName'@'localhost';
grant all privileges on webdb.* to 'web'@'localhost'; revoke all on webdb.* from 'wdp'@'localhost';
use webdb;
```

### 10.2 도서정보 테이블 생성 및 데이터 입력

[goodsinfo] 테이블 생성

```
create table goodsinfo(
    code char(4) not null,
    title varchar(50) not null,
    writer varchar(20),
    price int(8) not null,
    primary key(code)
);
```

[goodsinfo] 테이블에 데이터 입력

```
insert into goodsinfo(code, title, writer, price) values('1001', '뇌를 자극하는 java 프로그래밍', '김윤령', 27000);
insert into goodsinfo(code, title, writer, price) values('1002', '좋은 사치를 만드는 노출', '정승익', 20000);
insert into goodsinfo(code, title, writer, price) values('1003', '예제로 배우는 엑셀 2007', '김대식', 19000);
insert into goodsinfo(code, title, writer, price) values('1004', '지금 당장 경제공부 시작하라', '최진기', 18500);
insert into goodsinfo(code, title, writer, price) values('1005', 'Introduction to Algorithms', '토마스', 42000);
insert into goodsinfo(code, title, writer, price) values('1006', '이것이 자바다.', '신용권', 32000);
```

### 10.3 JDBC 드라이버 다운로드 받기

- <https://dev.mysql.com/downloads/connector/>에서 Connector/J를 다운로드 받아 압축을 풀어준다.
- mysql-connector-java-5.1.10-bin.jar 파일을 톰캣의 lib 폴더나 [프로젝트]-[WebContent]-[WEB-INF]-[lib]에 복사한다.
- 톰캣이 실행 중이라면 재가동한다. (oracle인 경우 ojdbc6.jar 파일)

### 10.4 JDBC 드라이버를 로드하고, 데이터베이스로 연결하기

1) 데이터베이스 연결을 하기 위해서는 JDBC 드라이버를 로드한다.

- MySQL : `Class.forName("com.mysql.jdbc.Driver");`
- Oracle : `Class.forName("oracle.jdbc.driver.OracleDriver")`

2) 데이터베이스를 연결한다.

- MySQL: `DriverManager.getConnection("jdbc:mysql://localhost:3306/webdb", "web", "pass")`
- Oracle: `DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "web", "pass")`

## 10.5 사용자 관리 프로그램 (CURD)

[userinfo] 테이블 생성

```
create table userinfo (
    name varchar(10) not null,
    id varchar(8) not null,
    password varchar(8) not null,
    primary key(id)
);
```

[userinfo] Sample 데이터 입력

```
insert into userinfo(name, id, password) values('김재영', 'spider', 'pass');
insert into userinfo(name, id, password) values('박철규', 'iceman', 'pass');
insert into userinfo(name, id, password) values('변재희', 'sparrow', 'pass');
insert into userinfo(name, id, password) values('진석영', 'batman', 'pass');
```

[java Resources]-[src]-[user]-[UserVO.java]

```
public class UserVO {
    private String id;
    private String name;
    private String password;
    // ... getter()/setter()/toString()
}
```

### [사용자 목록]

[사용자등록]

아이디	이름	비밀번호
iceman	김철규	pass
sparrow	변재희	pass
spider	김재영	pass

### [사용자 등록]

아이디	<input type="text"/>
비밀번호	<input type="password"/>
이름	<input type="text"/>

### [사용자 정보]

아이디	<input type="text" value="sparrow"/>
비밀번호	<input type="password" value="...."/>
이름	<input type="text" value="변재희"/>

[java Resources]-[src]-[user]-[UserDAO.java]

```
public class UserDAO {
    //데이터베이스 접속 -----//
    public Connection con(){
        Connection CON=null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            CON=DriverManager.getConnection("jdbc:mysql://localhost:3306/webdb", "root", "1234");
            System.out.println("접속성공");
        }catch(Exception e) {
            System.out.println("접속실패:" + e.toString());
        }
        return CON;
    }
}
```

```

public ArrayList<UserVO> list(){
    ArrayList<UserVO> array = new ArrayList<UserVO>();
    try{
        Connection con=con();
        String sql = "select * from userinfo";
        PreparedStatement ps=con.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        while(rs.next()){
            UserVO vo = new UserVO();
            vo.setId(rs.getString("id"));
            vo.setName(rs.getString("name"));
            vo.setPassword(rs.getString("password"));
            array.add(vo);
        }
    }catch(Exception e){ System.out.println(e.toString()); }
    return array;
}

public void insert(UserVO vo){
    try{
        Connection con=con();
        String sql = "insert into userinfo(id, name, password) values(?, ?, ?)";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, vo.getId());
        ps.setString(2, vo.getName());
        ps.setString(3, vo.getPassword());
        ps.execute();
    }catch(Exception e){
        System.out.println(e.toString());
    }
}

public UserVO read(String id){
    UserVO vo = new UserVO();
    try{
        Connection con=con();
        String sql = "select * from userinfo where id=?";
        PreparedStatement ps=con.prepareStatement(sql);
        ps.setString(1, id);
        ResultSet rs=ps.executeQuery();
        if(rs.next()){
            vo.setId(rs.getString("id"));
            vo.setName(rs.getString("name"));
            vo.setPassword(rs.getString("password"));
        }
    }catch(Exception e){ System.out.println(e.toString()); }
    return vo;
}

public void delete(String id){
    try{
        Connection con=con();
        String sql = "delete from userinfo where id=?";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, id);
        ps.execute();
    }catch(Exception e){ System.out.println(e.toString()); }
}

public void update(UserVO vo){
    try{
        Connection con=con();
        String sql = "update userinfo set name=?, password=? where id=?";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, vo.getName());
        ps.setString(2, vo.getPassword());
        ps.setString(3, vo.getId());
        ps.execute();
    }catch(Exception e){ System.out.println(e.toString()); }
}
}

```

[java Resources]-[src]-[user]-[UserServlet.java]

```
@WebServlet( value={"/user/list", "/user/insert", "/user/read", "/user/update", "/user/delete"})
public class UserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        UserDao dao = new UserDao();
        switch(request.getServletPath()){
            case "/user/list":
                ArrayList<UserVO> array=dao.list();
                request.setAttribute("list", array);
                RequestDispatcher dis=request.getRequestDispatcher("list.jsp");
                dis.forward(request, response);
                break;
            case "/user/insert":
                response.sendRedirect("insert.jsp");
                break;
            case "/user/read":
                String id=request.getParameter("id");
                UserVO vo = dao.read(id);
                request.setAttribute("vo", vo);
                dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        UserDao dao = new UserDao();
        UserVO vo = new UserVO();
        vo.setId(request.getParameter("id"));
        vo.setName(request.getParameter("name"));
        vo.setPassword(request.getParameter("password"));
        switch (request.getServletPath()) {
            case "/user/insert":
                dao.insert(vo);
                response.sendRedirect("list");
                break;
            case "/user/update":
                dao.update(vo);
                response.sendRedirect("list");
                break;
            case "/user/delete":
                dao.delete(vo.getId());
                response.sendRedirect("list");
        }
    }
}
```

[WebContent]-[user]-[list.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<head>
    <title>사용자 목록</title>
    <link rel="stylesheet" href="../../css/home.css">
</head>
<body>
    <h1>[사용자 목록]</h1>
    <a href="insert">[사용자등록]</a>
    <table>
        <tr class="title">
            <td width=100>아이디</td><td width=200>이름</td><td width=100>비밀번호</td>
        </tr>
        <c:forEach items="${list}" var="vo">
            <tr class="row" onClick="location.href='read?id=${vo.id}'">
                <td>${vo.id}</td><td>${vo.name}</td><td>${vo.password}</td>
            </tr>
        </c:forEach>
    </table>
</body>
</html>
```

[WebContent]-[css]-[home.css]

```
@CHARSET "UTF-8";
table {border-collapse:collapse;}
.title {background:gray; color:white; text-align:center;}
.row:hover {background:gray; color:white; cursor:pointer;}
td {height:30px; border:solid 1px black; padding:5px;}
```

[WebContent]-[user]-[insert.jsp]

```
...
<body>
  <h1>[사용자 등록]</h1>
  <form name="frm" action="insert" method="post">
    <table>
      <tr><td class="title" width=100>아이디</td><td width=300><input type="text" name="id"></td></tr>
      <tr><td class="title">비밀번호</td><td><input type="password" name="password"></td></tr>
      <tr><td class="title">이름</td><td><input type="text" name="name"></td></tr>
    </table>
    <input type="submit" value="저장">
    <input type="reset" value="취소">
    <input type="button" value="목록" onClick="location.href='list'">
  </form>
</body>
</html>
```

[WebContent]-[user]-[read.jsp]

```
...
<body>
  <h1>[사용자 정보]</h1>
  <form name="frm">
    <table>
      <tr>
        <td class="title" width=100>아이디</td>
        <td width=300><input type="text" name="id" value="{vo.id}" readonly></td>
      </tr>
      <tr>
        <td class="title">비밀번호</td>
        <td><input type="password" name="password" value="{vo.password}"></td>
      </tr>
      <tr>
        <td class="title">이름</td>
        <td><input type="text" name="name" value="{vo.name}"></td>
      </tr>
    </table>
    <input type="submit" value="저장">
    <input type="reset" value="취소">
    <input type="button" value="목록" onClick="location.href='list'">
    <input type="button" value="삭제" id="btnDelete">
  </form>
</body>
<script>
  $(frm).submit(function(event){
    event.preventDefault();
    if(!confirm("수정하실래요?")) return;

    frm.action = "update";
    frm.method = "post";
    frm.submit();
  });

  $("#btnDelete").on("click", function(){
    if(!confirm("삭제하실래요?")) return;

    frm.action="delete";
    frm.method = "post";
    frm.submit();
  })
</script>
</html>
```



## 10.6 사용자 관리 프로그램 Ajax 처리

Ajax은 비동기식 자바스크립트 XML(Asynchronous Javascript And XML)의 약자이다. HTML 만으로 어려운 다양한 작업을 웹페이지에서 구현해 이용자가 웹페이지와 자유롭게 상호 작용할 수 있도록 하는 기술이다. 별도 프로그램을 설치하거나 웹페이지를 다시 로딩하지 않고도 메뉴 등 화면상의 객체를 자유롭게 움직이고 다룰 수 있다. 액티브X나 플래시 등에 비해 가볍고 속도가 빨라 차세대 웹 기술로 각광받고 있다.

- JSON 데이터 생성: [WebContent]-[WEB-INF]-[lid] 폴더에 json-simple-x.x.x.jar 파일을 다운로드 받아 복사한다.

```
[java Resources]-[src]-[user]-[JSONServlet.java]
```

```
@WebServlet(value={"/user/", "/user/list.json", "/user/insert.json", "/user/delete.json", "/user/update.json"})
public class JSONServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        UserDAO dao = new UserDAO();

        switch(request.getServletPath()){
            case "/user":
                response.sendRedirect("user.jsp");
                break;

            case "/user/list.json":
                ArrayList<UserVO> users = dao.list();
                JSONArray jArray = new JSONArray();

                for(UserVO user:users){
                    JSONObject obj=new JSONObject();
                    obj.put("id", user.getId());
                    obj.put("name", user.getName());
                    obj.put("password", user.getPassword());
                    jArray.add(obj);
                }

                out.println(jArray.toJSONString());
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        UserDAO dao = new UserDAO();

        UserVO vo = new UserVO();
        vo.setId(request.getParameter("id"));
        vo.setName(request.getParameter("name"));
        vo.setPassword(request.getParameter("password"));

        switch (request.getServletPath()) {
            case "/user/insert.json":
                dao.insert(vo);
                break;

            case "/user/update.json":
                dao.update(vo);
                break;

            case "/user/delete.json":
                dao.delete(vo.getId());
                break;
        }
    }
}
```

## [사용자 등록]

아이디	<input type="text" value="hd9536"/>
비밀번호	<input type="password" value="****"/>
이름	<input type="text"/>

## [사용자 목록]

<input type="checkbox"/>	아이디	이름	비밀번호	수정	삭제
<input type="checkbox"/>	iceman	<input type="text" value="김철규"/>	<input type="text" value="pass"/>	<input type="button" value="수정"/>	<input type="button" value="삭제"/>
<input type="checkbox"/>	sparrow	<input type="text" value="변재희"/>	<input type="text" value="pass"/>	<input type="button" value="수정"/>	<input type="button" value="삭제"/>
<input type="checkbox"/>	spider	<input type="text" value="김재영"/>	<input type="text" value="pass"/>	<input type="button" value="수정"/>	<input type="button" value="삭제"/>

[WebContent]-[user]-[user.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
  <title>사용자 목록</title>
  <style>
    table {border-collapse:collapse;}
    .title {background:gray; color:white; text-align:center;}
    .row:hover {background:gray; color:white; cursor:pointer;}
    td {height:30px; border:solid 1px black; padding:5px;}
  </style>
</head>
<body>
  <h1>[사용자 등록]</h1>
  <table>
    <tr>
      <td class="title" width=100>아이디</td>
      <td width=350><input type="text" id="id"></td>
    </tr>
    <tr>
      <td class="title">비밀번호</td>
      <td><input type="password" id="password"></td>
    </tr>
    <tr>
      <td class="title">이름</td>
      <td><input type="text" id="name"></td>
    </tr>
  </table>
  <input type="button" value="저장" id="btnInsert">

  <h1>[사용자 목록]</h1>
  <button id="btnChkDelete">선택삭제</button>
  <table id="tbl"></table>
</body>
<script>
  getList();
  function getList(){
    var str = "<tr class=title>";
    str += "<td><input type=checkbox id='chkAll'></td>";
    str += "<td width=100>아이디</td>";
    str += "<td width=100>이름</td>";
    str += "<td width=100>비밀번호</td>";
    str += "<td width=50>수정</td>";
    str += "<td width=50>삭제</td>";
    str += "</tr>";
  }
</script>
```

```

$.ajax({
    type:"get",
    url:"list.json",
    dataType:"json",
    success:function(data){
        $(data).each(function(){
            var id=this.id;
            var name=this.name;
            var password=this.password;
            str += "<tr class=row>";
            str += "<td><input type=checkbox class=chk></td>";
            str += "<td class=id>" + id + "</td>";
            str += "<td><input type=text class=name size=5 value='" + name + "'></td>";
            str += "<td><input type=text class=password size=5 value='" + password + "'></td>";
            str += "<td><button class=btnUpdate>수정</button></td>";
            str += "<td><button class=btnDelete>삭제</td>";
            str += "</tr>";
        });
        $("#tbl").html(str);
    }
});

$("#chkAll").prop("checked", false);
$("#tbl .chk").each(function(){
    $(this).prop("checked", false);
});
}

$("#btnInsert").on("click", function(){
    var id = $("#id").val();
    var name = $("#name").val();
    var password = $("#password").val();
    $.ajax({
        type:"post",
        url:"insert.json",
        data:{'id':id, 'name':name, 'password':password},
        success:function(){
            alert("저장되었습니다!"); getList();
        }
    });
});

$("#tbl").on("click", ".btnUpdate", function(){
    var row = $(this).parent().parent();
    var id = row.find(".id").html();
    var name = row.find(".name").val();
    var password = row.find(".password").val();
    if(!confirm("수정하실래요?")) return;
    $.ajax({
        type:"post",
        url:"update.json",
        data:{'id':id, 'name':name, 'password':password},
        success:function(){
            alert("수정되었습니다!"); getList();
        }
    });
});

$("#tbl").on("click", ".btnDelete", function(){
    var row = $(this).parent().parent();
    var id = row.find(".id").html();
    if(!confirm("삭제하실래요?")) return;
    $.ajax({
        type:"post",
        url:"delete.json",
        data:{'id':id},
        success:function(){
            alert("삭제되었습니다!"); getList();
        }
    });
});
});

```

```

$("#tbl").on("click", "#chkAll", function(){
    if($("#tbl").is(":checked")){
        $("#tbl .chk").each(function(){ $(this).prop("checked", true); });
    }else{
        $("#tbl .chk").each(function(){ $(this).prop("checked", false); });
    }
});

$("#tbl").on("click", ".chk", function(){
    var isChecked = true;
    $("#tbl .chk").each(function(){
        if(!$(this).is(":checked")) isChecked = false;
    });
    if(isChecked){ $("#chkAll").prop("checked", true); }
    else{ $("#chkAll").prop("checked", false); }
});

$("#btnChkDelete").on("click", function(){
    if(!confirm("삭제하시겠습니까?")) return;
    $("#tbl .row .chk:checked").each(function(){
        var row = $(this).parent().parent();
        var id = row.find(".id").html();
        $.ajax({
            type:"post",
            url:"delete.json",
            data:{"id":id},
            success:function(){ }
        });
    });
    alert("삭제되었습니다!");
    getList();
});
</script>
</html>

```

## • Handlebars 이용

[WebContent] - [user] - [user.jsp]

```

<script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
...
<table id="tbl"></table>
<script id="tempForm" type="text/x-handlebars-template">
...
{{#each .}}
<tr class="row">
    <td><input type="checkbox" class="chk"></td>
    <td class="id">{{id}}</td>
    <td><input type="text" class="name" value="{{name}}" size=5></td>
    <td><input type="text" class="password" value="{{password}}" size=5></td>
    <td><button class="btnUpdate">수정</button></td>
    <td><button class="btnDelete">삭제</button></td>
</tr>
{{/each}}
</script>
...
<script>
function getList(){
    $.ajax({
        type:"get",
        url:"list.json",
        dataType:"json",
        success:function(data){
            var template = Handlebars.compile($("#tempForm").html());
            $("#tbl").html(template(data));
        }
    });
}
</script>

```

- jdom.jar 파일을 이용한 XML 파일 생성

- [WebContent]-[WEB-INF]-[lib] 폴더에 jdom.jar 파일을 다운로드 받아 복사한다.

[java Resources]-[src]-[user]-[JSONServlet.java]

```
@WebServlet(value={"/user/", "/user/list.json", "/user/insert.json", "/user/delete.json", "/user/update.json", "/user/list.xml"})
```

```
public class JSONServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        UserDao dao = new UserDao();

        switch(request.getServletPath()){
            case "/user/list.xml":
                users = dao.list();
                try{
                    Document doc = new Document();
                    Element root = new Element("root");
                    doc.setContent(root);

                    for(UserVO user:users){
                        Element item = new Element("item");
                        root.addContent(item);

                        Element id = new Element("id");
                        id.setText(user.getId());
                        item.addContent(id);

                        Element name = new Element("name");
                        name.setText(user.getName());
                        item.addContent(name);

                        Element password = new Element("password");
                        password.setText(user.getPassword());
                        item.addContent(password);
                    }

                    String xml = new XMLOutputter().outputString(doc);
                    out.println(xml);
                }catch(Exception e){
                    System.out.println(e.toString());
                }
                break;
            ...
        }
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <item>
    <id>iceman</id>
    <name>김철규</name>
    <password>pass</password>
  </item>
  <item>
    <id>sparrow</id>
    <name>변재희</name>
    <password>pass</password>
  </item>
  <item>
    <id>spider</id>
    <name>김재영</name>
    <password>pass</password>
  </item>
</root>
```

[Webcontent]-[user]-[user.jsp] 추가

```
<script>
    getXML();

    function getXML(){
        $.ajax({
            type:"get",
            url:"list.xml",
            dataType:"xml",
            success:function(data){
                $(data).find("item").each(function(){
                    var id = $(this).find("id").text();
                    var name = $(this).find("name").text();
                    var password = $(this).find("password").text();
                    alert(id + "\n" + name + "\n" + password);
                });
            }
        });
    }
    ...
</script>
```

## 10.7 상품관리 프로그램 (CURD)

### [상품목록]

[상품등록]



### [상품등록]

### [상품정보]

상품코드	<input type="text"/>	상품코드	P001
상품명	<input type="text"/>	상품명	5단서랍장
상품가격	<input type="text"/> 원	상품가격	50000
상품이미지	<div>150 x 120</div>	상품이미지	

저장 취소 목록
수정 삭제 취소 목록

[productinfo] 테이블 생성

```
create table productinfo(
    code char(4) primary key,
    pname varchar(100) not null,
    price int,
    image varchar(50)
);
```

[productinfo] Sample 데이터 입력

```
insert into productinfo(code, pname, price, image) values('P001', '5단서랍장', 50000, 'img01.jpg');
insert into productinfo(code, pname, price, image) values('P002', '원목싱글침대', 800000, 'img02.jpg');
insert into productinfo(code, pname, price, image) values('P003', '싱글침대', 800000, 'img03.jpg');
insert into productinfo(code, pname, price, image) values('P004', '4인용 가죽 소파', 1800000, 'img04.jpg');
```

[Servers]-[server.xml]

```
<host>
  <Context docBase="c:/image" path="/product/img" reloadable="true"/>
</host>
```

[java Resources]-[src]-[[Database.java]

```
public class Database {
    public static Connection CON;
    static {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            CON=DriverManager.getConnection("jdbc:mysql://localhost:3306/webdb", "root", "1234");
        }catch(Exception e) { System.out.println("접속실패:" + e.toString()); }
    }
}
```

cos.jar 파일을 [프로젝트]-[WEB-INF]-[lib] 폴더에 복사한다.

[java Resources]-[src]-[product]-[ProductVO.java]

```
public class ProductVO {
    private String code;
    private String pname;
    private int price;
    private String image;
    //... setter()/getter()/toString()
}
```

```

public class ProductDAO {
    public ArrayList<ProductVO> list() {
        ArrayList<ProductVO> array = new ArrayList<ProductVO>();
        try {
            String sql = "select * from productinfo";
            PreparedStatement ps = Database.CON.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                ProductVO vo = new ProductVO();
                vo.setCode(rs.getString("code"));
                vo.setPname(rs.getString("pname"));
                vo.setPrice(rs.getInt("price"));
                vo.setImage(rs.getString("image"));
                array.add(vo);
            }
        } catch (Exception e) { System.out.println(e.toString()); }
        return array;
    }

    public void insert(ProductVO vo) {
        try {
            String sql = "insert into productinfo(code, pname, price, image) values(?, ?, ?, ?)";
            PreparedStatement ps = Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getCode());
            ps.setString(2, vo.getPname());
            ps.setInt(3, vo.getPrice());
            ps.setString(4, vo.getImage());
            ps.execute();
        } catch (Exception e) { System.out.println(e.toString()); }
    }

    public ProductVO read(String code) {
        ProductVO vo = new ProductVO();
        try {
            String sql = "select * from productinfo where code=?";
            PreparedStatement ps = Database.CON.prepareStatement(sql);
            ps.setString(1, code);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                vo.setCode(rs.getString("code"));
                vo.setPname(rs.getString("pname"));
                vo.setPrice(rs.getInt("price"));
                vo.setImage(rs.getString("image"));
            }
        } catch (Exception e) { System.out.println(e.toString()); }
        return vo;
    }

    public void delete(String code) {
        try {
            String sql = "delete from productinfo where code=?";
            PreparedStatement ps = Database.CON.prepareStatement(sql);
            ps.setString(1, code);
            ps.execute();
        } catch (Exception e) { System.out.println(e.toString()); }
    }

    public void update(ProductVO vo) {
        try {
            String sql = "update productinfo set pname=?, price=?, image=? where code=?";
            PreparedStatement ps = Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getPname());
            ps.setInt(2, vo.getPrice());
            ps.setString(3, vo.getImage());
            ps.setString(4, vo.getCode());
            ps.execute();
        } catch (Exception e) { System.out.println(e.toString()); }
    }
}

```

[java Resources]-[src]-[product]-[ProductServlet.java]

```
@WebServlet(value={"/product/list", "/product/insert", "/product/read", "/product/delete", "/product/update"})
public class ProductServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ProductDAO dao = new ProductDAO();
        switch(request.getServletPath()){
            case "/product/list":
                request.setAttribute("list", dao.list());
                RequestDispatcher dis=request.getRequestDispatcher("list.jsp");
                dis.forward(request, response);
                break;
            case "/product/insert":
                dis=request.getRequestDispatcher("insert.jsp");
                dis.forward(request, response);
                break;
            case "/product/read":
                String code=request.getParameter("code");
                request.setAttribute("vo", dao.read(code));
                dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;
            case "/product/delete":
                code = request.getParameter("code");
                ProductVO vo=dao.read(code);
                String path="c:" + File.separator + "/image" + File.separator;
                if(vo.getImage()!=null){
                    String image=path + vo.getImage();
                    File file=new File(image);
                    file.delete();
                }
                dao.delete(code);
                response.sendRedirect("list");
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        ProductDAO dao=new ProductDAO();
        ProductVO vo=new ProductVO();
        String path="c:/image";
        File mdPath = new File(path);
        if(!mdPath.exists()) mdPath.mkdir();
        MultipartRequest multi = new MultipartRequest(
            request, path, 1024 * 1024 * 10, "UTF-8", new DefaultFileRenamePolicy()
        );
        vo.setCode(multi.getParameter("code"));
        vo.setPname(multi.getParameter("pname"));
        vo.setPrice(Integer.parseInt(multi.getParameter("price")));
        vo.setImage(multi.getFilesystemName("image"));

        switch(request.getServletPath()){
            case "/product/insert":
                dao.insert(vo);
                response.sendRedirect("list");
                break;
            case "/product/update":
                ProductVO oldVO = dao.read(vo.getCode());
                if(multi.getFilesystemName("image") == null){
                    vo.setImage(oldVO.getImage());
                }else{
                    if(oldVO.getImage() != null){
                        String image=path + File.separator + oldVO.getImage();
                        File file=new File(image);
                        file.delete();
                    }
                }
                dao.update(vo);
                response.sendRedirect("list");
        }
    }
}
```



#### [WebContet]-[product]-[list.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<html>
<head>
    <style>
        #container {width:680px;background:gray; padding:10px; overflow:hidden;}
        .box {width:150px; background:pink; padding:5px; margin:5px; float:left;}
    </style>
</head>
<body>
    <h1>[상품목록]</h1>
    <a href="insert">[상품등록]</a>
    <div id="container">
    <c:forEach items="${list}" var="vo">
        <div class="box">
            
            <div>${vo.code}</div><div>${vo.pname}</div><div>${vo.price}원</div>
        </div>
    </c:forEach>
    </div>
</body>
</html>
```

#### [WebContet]-[product]-[insert.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
    <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
    <title>상품등록</title>
    <style>
        table {border-collapse:collapse;}
        td {border:solid 1px black; height:30px;}
        .title {background:gray; color:white; text-align:center;}
    </style>
</head>
<body>
    <h1>[상품등록]</h1>
    <form name="frm" action="insert" method="post" enctype="multipart/form-data">
        <table>
            <tr><td width=100 class="title">상품코드</td><td width=500><input type="text" name="code"></td></tr>
            <tr><td width=100 class="title">상품명</td><td width=500><input type="text" name="pname" size=50></td></tr>
            <tr><td width=100 class="title">상품가격</td><td width=500><input type="text" name="price">원</td></tr>
            <tr>
                <td width=100 class="title">상품이미지</td>
                <td width=500>
                    
                    <input type="file" name="image" accept="image/*" style="visibility:hidden;">
                </td>
            </tr>
        </table>
        <input type="submit" value="저장">
        <input type="reset" value="취소">
        <input type="button" value="목록" onClick="location.href='list'">
    </form>
</body>
<script>
    $("#image").on("click", function(){
        $(frm.image).click();
    });

    $(frm.image).on("change", function(e){
        var reader = new FileReader();
        reader.readAsDataURL(this.files[0]);
        reader.onload=function(e){
            $("#image").attr("src", e.target.result);
        }
    });
</script>
</html>
```

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
  <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
  <title>상품정보</title>
  <style>
    table {border-collapse:collapse;}
    td {border:solid 1px black; height:30px;}
    .title {background:gray; color:white; text-align:center;}
  </style>
</head>
<body>
  <h1>[상품정보]</h1>
  <form name="frm" action="update" method="post" enctype="multipart/form-data">
    <table>
      <tr>
        <td width=100 class="title">상품코드</td>
        <td width=500><input type="text" name="code" value="{vo.code}"></td>
      </tr>
      <tr>
        <td width=100 class="title">상품명</td>
        <td width=500><input type="text" name="pname" value="{vo.pname}"></td>
      </tr>
      <tr>
        <td width=100 class="title">상품가격</td>
        <td width=500><input type="text" name="price" value="{vo.price}"></td>
      </tr>
      <tr>
        <td width=100 class="title">상품이미지</td>
        <td width=500>
          
          <input type="file" name="image" accept="image/*" style="visibility:hidden;">
        </td>
      </tr>
    </table>
    <input type="submit" value="수정">
    <input type="button" value="삭제" id="btnDelete">
    <input type="reset" value="취소">
    <input type="button" value="목록" onClick="location.href='list'">
  </form>
</body>
<script>
  $("#image").on("click", function(){
    $(frm.image).click();
  });

  $(frm.image).on("change", function(e){
    var reader = new FileReader();
    reader.onload=function(e){
      $("#image").attr("src", e.target.result);
    }
    reader.readAsDataURL(this.files[0]);
  });

  $(frm).on("submit", function(e){
    e.preventDefault();
    if(!confirm("수정하실래요?")) return;
    frm.submit();
  });

  $("#btnDelete").on("click", function(){
    if(!confirm("삭제하실래요?")) return;
    frm.method="get";
    frm.action="delete";
    frm.submit();
  });
</script>
</html>

```

## 10.8 상품관리 입력 프로그램 (다중이미지 업로드)

- 다중업로드를 위한 라이브러리(commons-fileupload.jar, commons-io.jar)를 [WebContent]-[WEB-INF]-[lib]에 복사한다.

[images] 테이블 생성

```
create table images(  
    image varchar(200),  
    code char(4),  
    primary key(image, code),  
    foreign key(code) references productinfo(code)  
);
```

### [상품등록]

상품코드	<input type="text"/>
상품명	<input type="text"/>
상품가격	<input type="text"/> 원
상품이미지	
첨부이미지	  
<input type="button" value="저장"/> <input type="button" value="취소"/> <input type="button" value="목록"/>	

[WebContet]-[product]-[insert.jsp]

```
<head>  
    <style>  
        ...  
        input[type='file']{visibility:hidden; height:0px;}  
        #listFile img {width:130px;margin:10px;}  
    </style>  
</head>  
<body>  
    <h1>[상품등록]</h1>  
    <form name="frm" action="insert" method="post" enctype="multipart/form-data">  
        <table>  
            ...  
            <tr>  
                <td width=100 class="title"><input type="button" value="첨부이미지" id="btnImage"></td>  
                <td style="height:150px;">  
                    <input type="file" name="images" accept="image/*" multiple>  
                    <div id="listFile" style="width:500px;"></div>  
                </td>  
            </tr>  
        </table>  
        <input type="submit" value="저장"><input type="reset" value="취소">  
        <input type="button" value="목록" onClick="location.href='list'">  
    </form>  
</body>  
<script>  
    ...  
    $("#btnImage").on("click", function(){  
        $(frm.images).click();  
    });  
  
    $(frm.images).change(function(){  
        var target = $(frm.images);  
        var html = "";  
        $.each(target[0].files, function(index, file){  
            html += "<img src='" + URL.createObjectURL(file) + "'>";  
        });  
        $("#listFile").html(html);  
    });  
</script>
```

[java Resources]-[src]-[product]-[ProductDAO.java]

```
public void insertImages(ArrayList<String> images, String code){
    try{
        for(String image:images){
            String sql="insert into images(image, code) values(?, ?)";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, image);
            ps.setString(2, code);
            ps.execute();
        }
    }catch(Exception e){ System.out.println(e.toString()); }
}
```

[java Resources]-[src]-[product]-[ProductServlet.java]

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    ProductDAO dao=new ProductDAO();
    ProductVO vo=new ProductVO();

    //업로드할 첨부이미지 저장변수
    ArrayList<String> images=new ArrayList<String>();

    String path="c:" + File.separator + "image" + File.separator;
    File mdPath = new File(path);
    if(!mdPath.exists()) mdPath.mkdir();

    DiskFileItemFactory fatory = new DiskFileItemFactory(); //1.업로드 데이터를 보관하는 FileItem의 Factory 설정
    ServletFileUpload upload = new ServletFileUpload(fatory); //2.업로드 요청을 처리하는 ServletFileUpload 생성
    upload.setHeaderEncoding("UTF-8"); //3.파일명 한글 처리

    try{
        //4.업로드 요청시 파싱해서 FileItem 목록 구함
        List<FileItem> items = upload.parseRequest(request);
        for(FileItem item:items){
            if(item.isFormField()){ //폼필드인 경우
                switch(item.getFieldName()){
                    case "code":
                        vo.setCode(item.getString("UTF-8")); break;
                    case "pname":
                        vo.setPname(item.getString("UTF-8")); break;
                    case "price":
                        vo.setPrice(Integer.parseInt(item.getString("UTF-8"))); break;
                }
            }else{ //업로드파일인 경우
                String fileName = item.getName();
                if(!fileName.equals("")){
                    if(item.getFieldName().equals("image")){
                        String newFileName = vo.getCode() + fileName.substring(fileName.lastIndexOf(".");
                        File file=new File(path + newFileName);
                        item.write(file);
                        vo.setImage(newFileName);
                    }else if(item.getFieldName().equals("images")){
                        String pathAttach = path + File.separator + vo.getCode() + File.separator;
                        mdPath = new File(pathAttach);
                        if(!mdPath.exists()) mdPath.mkdir();
                        File file=new File(pathAttach + fileName);
                        item.write(file);
                        images.add(fileName);
                    }
                }
            }
        }
    }catch(Exception e){ System.out.println(e.toString()); }

    switch(request.getServletPath()){
        case "/product/insert":
            dao.insert(vo);
            dao.insertImages(images, vo.getCode());
            response.sendRedirect("list");
            break;
        ...
    }
}
```

## 10.9 상품관리 읽기, 수정, 삭제 프로그램 (다중이미지 업로드)

### [상품정보]

상품코드	<input type="text" value="P005"/>
상품명	<input type="text" value="퀵침대"/>
상품가격	<input type="text" value="50000"/>
상품이미지	
첨부이미지	 

[WebContet]-[product]-[read.jsp]

```

<body>
  <h1>[상품정보]</h1>
  <form name="frm" action="update" method="post" enctype="multipart/form-data">
    <table>
      ...
      <tr>
        <td width=100 class="title">상품가격</td>
        <td width=500><input type="text" name="price" value="{vo.price}"></td>
      </tr>
      <tr>
        <td width=100 class="title">
          <input type="button" value="첨부이미지" id="btnImage">
        </td>
        <td style="height:150px;">
          <input type="file" name="images" accept="image/*" multiple>
          <div id="listFile" style="width:500px;">
            <c:forEach items="{images}" var="image">
              
            </c:forEach>
          </div>
        </td>
      </tr>
    </table>
    <input type="submit" value="수정">
    <input type="button" value="삭제" id="btnDelete">
    <input type="reset" value="취소">
    <input type="button" value="목록" onClick="location.href='list'">
  </form>
</body>
<script>
  $(frm).on("submit", function(e){
    e.preventDefault();
    if(!confirm("수정하실래요?")) return;
    frm.submit();
  });

  $("#btnDelete").on("click", function(){
    if(!confirm("삭제하실래요?")) return;
    frm.method="get";
    frm.action="delete";
    frm.submit();
  });
  ...
</script>

```

[java Resources]-[src]-[product]-[ProductDAO.java]

```
public ArrayList<String> readImages(String code) {
    ArrayList<String> list=new ArrayList<String>();
    try {
        String sql="select * from images where code=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, code);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            list.add(rs.getString("image"));
        }
    }catch(Exception e) { System.out.println(e.toString()); }
    return list;
}

public void deleteImages(String code) {
    try {
        String sql="delete from images where code=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, code);
        ps.execute();
    }catch(Exception e) { System.out.println(e.toString()); }
}
```

[java Resources]-[src]-[product]-[ProductServlet.java]

```
@WebServlet(value={"/product/list", "/product/insert", "/product/read", "/product/delete", "/product/update"})
public class ProductServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ProductDAO dao = new ProductDAO();

        switch(request.getServletPath()){
            ...
            case "/product/read":
                String code=request.getParameter("code");
                request.setAttribute("vo", dao.read(code));
                request.setAttribute("images", dao.readImages(code));
                dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;

            case "/product/delete":
                code = request.getParameter("code");
                ProductVO vo=dao.read(code);
                String path="c:" + File.separator + "image" + File.separator;

                File delPath = new File(path + code); //첨부이미지 삭제
                if(delPath.exists()){
                    File[] files=delPath.listFiles();
                    for(int i=0; i<files.length; i++){
                        System.gc();
                        files[i].delete();
                    }
                    delPath.delete();
                    dao.deleteImages(code);
                }

                if(vo.getImage()!=null){ //대표이미지 삭제
                    System.gc();
                    String image=path + vo.getImage();
                    File file=new File(image);
                    file.delete();
                }

                dao.delete(code);
                response.sendRedirect("list");
                break;
            }
        }
    }
}
```

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    ProductDAO dao=new ProductDAO();
    ProductVO vo=new ProductVO();
    ...
    try{
        List<FormItem> items = upload.parseRequest(request);
        for(FileItem item:items){
            if(item.isFormField()){ //폼필드인 경우
                switch(item.getFieldName()){
                    case "code":
                        vo.setCode(item.getString("UTF-8"));
                        break;
                    case "pname":
                        vo.setPname(item.getString("UTF-8"));
                        break;
                    case "price":
                        vo.setPrice(Integer.parseInt(item.getString("UTF-8")));
                        break;
                }
            }else{ //업로드파일인 경우
                String fileName = item.getName();
                if(!fileName.equals("")){
                    if(item.getFieldName().equals("image")){
                        newFileName = vo.getCode() + fileName.substring(fileName.lastIndexOf(".")).replace(".", "");
                        File file=new File(path + newFileName);
                        if(file.exists()) {
                            File delFile=new File(path + newFileName);
                            delFile.delete();
                        }
                        item.write(file);
                        vo.setImage(newFileName);
                    }else if(item.getFieldName().equals("images")){
                        String pathAttach = path + File.separator + vo.getCode() + File.separator;
                        mdPath = new File(pathAttach);
                        if(!mdPath.exists()) {
                            mdPath.mkdir();
                        }else if(images.size()==0){ //첨부파일들수정인 경우 기존파일삭제
                            File[] files=mdPath.listFiles();
                            for(int i=0; i<files.length; i++){
                                System.gc();
                                files[i].delete();
                            }
                        }
                        File file=new File(pathAttach + fileName);
                        item.write(file);
                        images.add(fileName);
                    }
                }
            }
        }
    }catch(Exception e){ System.out.println(e.toString()); }

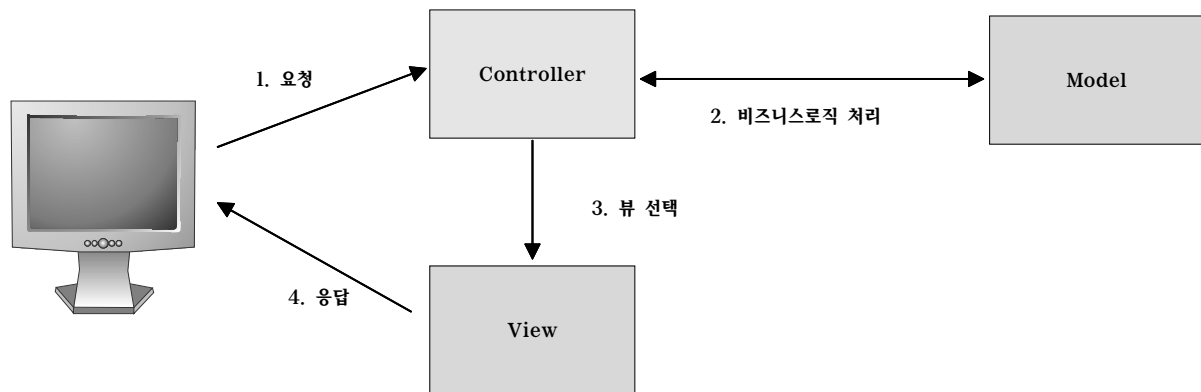
    switch(request.getServletPath()){
        case "/product/insert":
            dao.insert(vo);
            dao.insertImages(images, vo.getCode());
            response.sendRedirect("list");
            break;
        case "/product/update":
            ProductVO oldVO = dao.read(vo.getCode());
            if(newFileName == null) vo.setImage(oldVO.getImage()); //대표이미지가 변경된 경우
            if(images.size()>0){ //첨부이미지들이 변경된 경우
                dao.deleteImages(vo.getCode());
                dao.insertImages(images, vo.getCode());
            }
            dao.update(vo);
            response.sendRedirect("list");
        }
    }
}

```

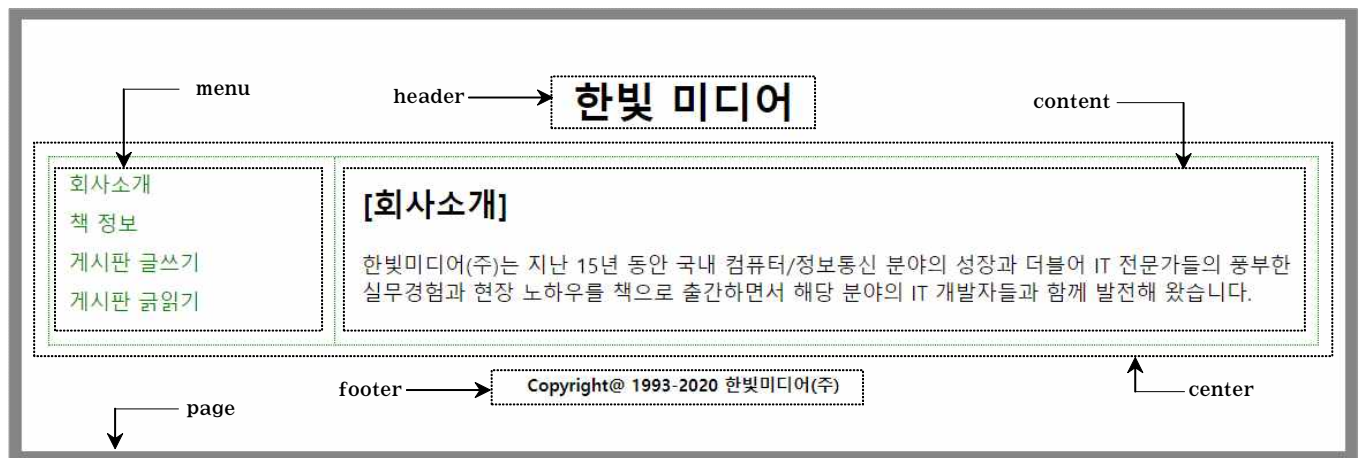
## 제11장 MVC모델 프로젝트 생성

### 11.1 MVC 모델 (Model-View-Controller)

- 모델(Model): 처리로직을 담당하는 부분으로 DB를 연동을 통해 데이터를 가져와 어떤 작업을 처리한 후 작업결과를 DB에 저장한다. DB연동 로직인 자바빈(Java Bean)과 처리 로직인 자바 클래스(Java Class)가 모델에 해당한다.
- 뷰(View): 뷰는 화면에 내용을 표시하는 역할을 담당하며 JSP가 뷰에 해당한다.
- 컨트롤러(Controller): 뷰와 모델 사이에서 이들의 흐름을 제어하는 역할을 담당한다. 컨트롤러는 사용자의 요청을 받아서 모델에 넘겨주고 모델이 처리한 작업의 결과를 다시 받아 뷰에 보내주는 역할을 한다. 보통 서블릿(Servlet)을 컨트롤러로 사용한다.



### 11.2 웹 템플릿 만들기



[webContent]-[header.jsp]

```
<h1>한빛 미디어</h1>
```

[webContent]-[footer.jsp]

```
<h3>Copyright@ 1993-2020 한빛미디어(주)</h3>
```

[webContent]-[menu.jsp]

```
<div class="item"><a href="/index.jsp">회사소개</a></div>
<div class="item"><a href="/book/list">책 정보</a></div>
<div class="item"><a href="/bbs/insert">게시판 글쓰기</a></div>
<div class="item"><a href="/bbs/list">게시판 글읽기</a></div>
```



#### [webContent]-[index.jsp]

```
<html>
<head>
  <title>한빛미디어</title>
  <link rel="stylesheet" href="home.css"/>
</head>
<body>
  <div id="page">
    <div id="header"><jsp:include page="header.jsp"/></div>
    <div id="center">
      <div id="menu">
        <jsp:include page="menu.jsp"/>
      </div>
      <div id="content">

        <!-- 여기에 출력할 내용 작성 시작 -->
        <h2>[회사소개]</h2>
        한빛미디어(주)는 지난 15년 동안 국내 컴퓨터/정보통신 분야의 성장과 더불어 IT 전문가들의
        풍부한 실무경험과 현장 노하우를 책으로 출간하면서 해당 분야의 IT 개발자들과 함께 발전해 왔습니다.
        <!-- 여기에 출력할 내용 작성 종료 -->

      </div>
    </div>
    <div id="footer"><jsp:include page="footer.jsp"/></div>
  </div>
</body>
</html>
```

#### [webContent]-[home.css]

```
@charset "UTF-8";
body{background:gray;}
#header, #footer {text-align:center;}
#page {background:white;width:980px; box-shadow:10px 10px 10px gray; margin:auto;padding:20px;}
#center{overflow:hidden; border:1px dotted green;}
#menu{width:200px; float:left; padding:10px;}
#content{width:720px; float:left; padding:0px 10px 30px 20px;; border-left:1px dotted green;}
.item {height:30px;} .item a{text-decoration:none; color:green;padding:5px;} .item a:hover{color:white; background:green;}
table {border-collapse:collapse;} td{border:1px solid green;}
#btnMenu{margin-top:5px;}
```

#### • JDBC 드라이버의 로드, 데이터베이스로 연결하기 및 테스트

##### [Java Resources]-[src]-[book]-[DataBase.java]

```
public class DataBase{
  public static Connection CON;
  static {
    try {
      Class.forName("com.mysql.jdbc.Driver");
      CON=DriverManager.getConnection("jdbc:mysql://localhost:3306/webdb", "root", "1234");
      System.out.println("접속성공");
    }catch(Exception e) {
      System.out.println("접속실패:" + e.toString());
    }
  }
}
```

##### [Java Resources]-[src]-[book]-[Main.java]

```
public class Main {
  public static void main(String[] args) throws Exception{
    System.out.println(Database.CON);
  }
}
```



## 11.3 도서 관리 프로그램 작성

[goodsinfo] 테이블 생성

```
create table goodsinfo(  
    code char(4) not null,  
    title varchar(50) not null,  
    writer varchar(20),  
    price int(8) not null,  
    primary key(code)  
);
```

[goodsinfo] 테이블에 데이터 입력

```
insert into goodsinfo(code, title, writer, price) values('1001', '뇌를 자극하는 java 프로그래밍', '김운명', 27000);  
insert into goodsinfo(code, title, writer, price) values('1002', '좋은 사지를 만드는 노출', '정승익', 20000);  
insert into goodsinfo(code, title, writer, price) values('1003', '예제로 배우는 엑셀 2007', '김대식', 19000);  
insert into goodsinfo(code, title, writer, price) values('1004', '지금 당장 경제공부 시작하라', '최진기', 18500);  
insert into goodsinfo(code, title, writer, price) values('1005', 'Introduction to Algorithms', '토마스', 42000);  
insert into goodsinfo(code, title, writer, price) values('1006', '이것이 자바다.', '신용권', 32000);  
insert into goodsinfo(code, title, writer, price) values('1007', '안드로이드 프로그래밍 정복1', '김상형', 35000);  
insert into goodsinfo(code, title, writer, price) values('1008', '안드로이드 프로그래밍 정복2', '김상형', 32000);  
insert into goodsinfo(code, title, writer, price) values('1009', '예제로 배우는 엑셀 2007', '김대식', 19000);  
insert into goodsinfo(code, title, writer, price) values('1010', '지금 당장 경제공부 시작하라', '최진기', 18500);  
insert into goodsinfo(code, title, writer, price) values('1011', 'Introduction to Algorithms', '토마스', 42000);  
insert into goodsinfo(code, title, writer, price) values('1012', '이것이 자바다.', '신용권', 32000);
```

[Java Resources]-[src]-[book]-[BookVO.java]

```
public class BBSVO {  
    private int seqno;  
    private String title;  
    private String content;  
    private String writer;  
    private Date wdate;  
    // ... setter()/getter()/toString() 추가  
}
```

[Java Resources]-[src]-[book]-[BookDAO.java]

```
public class BookDAO {  
    //도서 목록-----//  
    public ArrayList<BookVO> list(){  
        ArrayList<BookVO> array=new ArrayList<BookVO>();  
  
        try{  
            String sql="select * from goodsinfo";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ResultSet rs=ps.executeQuery();  
            while(rs.next()){  
                BookVO vo=new BookVO();  
                vo.setCode(rs.getString("code"));  
                vo.setTitle(rs.getString("title"));  
                vo.setPrice(rs.getInt("price"));  
                vo.setWriter(rs.getString("writer"));  
                array.add(vo);  
            }  
        }catch(Exception e) {  
            System.out.println(e.toString());  
        }  
        return array;  
    }  
  
    //도서 읽기 -----//
```

```

public BookVO read(String code){
    BookVO vo=new BookVO();
    try{
        String sql="select * from goodsinfo where code=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, code);
        ResultSet rs=ps.executeQuery();
        if(rs.next()){
            vo.setCode(rs.getString("code"));
            vo.setTitle(rs.getString("title"));
            vo.setPrice(rs.getInt("price"));
            vo.setWriter(rs.getString("writer"));
        }
    }catch(Exception e) {
        System.out.println(e.toString());
    }
    return vo;
}
//도서 입력-----//
public boolean insert(BookVO vo){
    try{
        String sql="insert into goodsinfo(code, title, writer, price) values(?, ?, ?, ?)";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getCode());
        ps.setString(2, vo.getTitle());
        ps.setString(3, vo.getWriter());
        ps.setInt(4, vo.getPrice());
        ps.execute();
        return true;
    }catch(Exception e){
        System.out.println(e.toString());
        return false;
    }
}
//도서 삭제-----//
public boolean delete(String code){
    try{
        String sql="delete from goodsinfo where code=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, code);
        ps.execute();
        return true;
    }catch(Exception e){
        System.out.println(e.toString());
        return false;
    }
}
//도서 수정-----//
public boolean update(BookVO vo){
    try{
        String sql="update goodsinfo set title=?, writer=?, price=? where code=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getTitle());
        ps.setString(2, vo.getWriter());
        ps.setInt(3, vo.getPrice());
        ps.setString(4, vo.getCode());
        ps.execute();
        return true;
    }catch(Exception e){
        System.out.println(e.toString());
        return false; }
}
}

```

```

@WebServlet(value= {"/book/list", "/book/insert", "/book/delete", "/book/update", "/book/read"})
public class BookServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        RequestDispatcher dis=null;
        BookDAO dao=new BookDAO();
        switch(request.getServletPath()){
            //도서 목록-----//
            case "/book/list":
                ArrayList<BookVO> list=new ArrayList<BookVO>();
                list=dao.list();
                request.setAttribute("list", list);
                dis=request.getRequestDispatcher("list.jsp");
                dis.forward(request, response);
                break;
            //도서 입력 GET-----//
            case "/book/insert":
                dis=request.getRequestDispatcher("insert.jsp");
                dis.forward(request, response);
                break;
            //도서 읽기-----//
            case "/book/read":
                String code=request.getParameter("code");
                BookVO vo=dao.read(code);
                request.setAttribute("vo", vo);
                dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;
            //도서 삭제-----//
            case "/book/delete":
                code=request.getParameter("code");
                dao.delete(code);
                response.sendRedirect("list");
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        BookDAO dao=new BookDAO();
        BookVO vo=new BookVO();
        switch(request.getServletPath()){
            //도서 입력 POST-----//
            case "/book/insert":
                dao.insert(vo);
                break;
            //도서 수정 -----//
            case "/book/update":
                vo.setCode(request.getParameter("code"));
                vo.setTitle(request.getParameter("title"));
                vo.setWriter(request.getParameter("writer"));
                String price=request.getParameter("price");
                vo.setPrice(Integer.parseInt(price));
                dao.update(vo);
                break;
        }
        response.sendRedirect("list");
    }
}

```

## [도서목록]

코드	제목	저자	가격
1001	뇌를 자극하는 java 프로그래밍	홍길동	32,000
1002	좋은 사지을 만드는 노출	정승익	20,000
1003	예제로 배우는 엑셀 2007	김대식	19,000
1004	지금 당장 경제공부 시작하라!	최진기	19,500
1005	Introduction to Algorithms	토마스 킴	42,000
1007	자바프로그래밍	홍길동	200

[webContent]-[book]-[list.jsp]

```
<h2>[도서목록]</h2>
<table border=1 width=700>
  <tr><td width=100>코드</td><td width=400>제목</td><td width=100>저자</td><td width=100>가격</td></tr>
  <c:forEach items="{list}" var="vo">
    <tr class="row">
      <td><a href="read?code={vo.code}">{vo.code}</a></td>
      <td>{vo.title}</td>
      <td>{vo.writer}</td>
      <td><fmt:formatNumber value="{vo.price}" pattern="#,###"/></td>
    </tr>
  </c:forEach>
</table>
```

## [도서등록]

코드:	<input type="text"/>
제목:	<input type="text"/>
저자:	<input type="text"/>
가격:	<input type="text"/>
저장	취소

## [도서수정]

코드:	<input type="text" value="1001"/>		
제목:	<input type="text" value="뇌를 자극하는 java 프로그래밍"/>		
저자:	<input type="text" value="홍길동"/>		
가격:	<input type="text" value="32000"/>		
저장	삭제	목록	취소

[webContent]-[book]-[insert.jsp]

```
<h2>[도서등록]</h2>
<form action="insert" method="post">
  <table border=1 width=500>
    <tr><td width=100>코드:</td><td><input type="text" name="code"></td></tr>
    <tr><td width=100>제목:</td><td><input type="text" name="title" size=50></td></tr>
    <tr><td width=100>저자:</td><td><input type="text" name="writer"></td></tr>
    <tr><td width=100>가격:</td><td><input type="text" name="price"></td></tr>
  </table>
  <div id="btnMenu"><input type="submit" value="저장"><input type="reset" value="취소"></div>
</form>
```

[webContent]-[book]-[read.jsp]

```
<h2>[도서수정]</h2>
<form action="update" method="post">
  <table border=1 width=500>
    <tr><td width=100>코드:</td><td><input type="text" name="code" value="{vo.code}"></td></tr>
    <tr><td width=100>제목:</td><td><input type="text" name="title" value="{vo.title}" size=50></td></tr>
    <tr><td width=100>저자:</td><td><input type="text" name="writer" value="{vo.writer}"></td></tr>
    <tr><td width=100>가격:</td><td><input type="text" name="price" value="{vo.price}"></td></tr>
  </table>
  <div id="btnMenu">
    <input type="submit" value="저장">
    <input type="button" value="삭제" onClick="location.href='delete?code={vo.code}'">
    <input type="button" value="목록" onClick="location.href='list'">
    <input type="reset" value="취소">
  </div>
</form>
```

## 11.4 게시판 관리 프로그램 작성

[bbs] 테이블 생성

```
create table bbs(
  seqno int(8) auto_increment,
  title varchar(50) not null,
  content varchar(500) not null,
  writer varchar(20) not null,
  wdate datetime not null default now(),
  primary key(seqno)
);
```

[bbs] 테이블에 데이터 입력

```
insert into bbs(title, content, writer) values('게시판 오픈', '많은 이용 부탁드립니다.', 'spider');
```

[Java Resources]-[src]-[book]-[BBSVO.java]

```
public class BBSVO {
  private int seqno;
  private String title;
  private String content;
  private String writer;
  private Date wdate;
  //... setter()/getter()/toString() 추가
}
```

[Java Resources]-[src]-[book]-[BBSDAO.java]

```
public class BBSDAO {
  //게시판 목록
  public ArrayList<BBSVO> list(int start){
    ArrayList<BBSVO> array=new ArrayList<BBSVO>();
    try{
      String sql="select * from bbs order by seqno desc limit ?, 5";
      PreparedStatement ps=Database.CON.prepareStatement(sql);
      ps.setInt(1, start);
      ResultSet rs=ps.executeQuery();
      while(rs.next()){
        BBSVO vo=new BBSVO();
        vo.setSeqno(rs.getInt("seqno"));
        vo.setTitle(rs.getString("title"));
        vo.setContent(rs.getString("content"));
        vo.setWriter(rs.getString("writer"));
        vo.setWdate(rs.getDate("wdate"));
        array.add(vo);
      }
    }catch(Exception e) {
      System.out.println(e.toString());
    }
    return array;
  }

  //게시판 글읽기
  public BBSVO read(int seqno){
    BBSVO vo= new BBSVO();
    try{
      String sql="select * from bbs where seqno=?";
      PreparedStatement ps=Database.CON.prepareStatement(sql);
      ps.setInt(1, seqno);
      ResultSet rs=ps.executeQuery();
      while(rs.next()){
        vo.setSeqno(rs.getInt("seqno"));
        vo.setTitle(rs.getString("title"));
        vo.setContent(rs.getString("content"));
        vo.setWriter(rs.getString("writer"));
        vo.setWdate(rs.getDate("wdate"));
      }
    }catch(Exception e) {
      System.out.println(e.toString());
    }
    return vo;
  }
}
```

```

//게시글 입력
public void insert(BBSVO vo) {
    try {
        String sql="insert into bbs(title, content, wdate, writer) values(?, ?, now(), ?)";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, vo.getTitle());
        ps.setString(2, vo.getContent());
        ps.setString(3, vo.getWriter());
        ps.execute();
    } catch (Exception e) { System.out.println(e.toString()); }
}

//게시글 갯수 출력
public int total() {
    int total=0;
    try {
        String sql="select count(seqno) total from bbs";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            total=rs.getInt("total");
        }
    } catch (Exception e) { System.out.println(e.toString()); }
    return total;
}
}

```

---

[Java Resources]-[src]-[book]-[BBServlet.java]

```

@WebServlet(value= {"/bbs/list", "/bbs/insert", "/bbs/read"})
public class BBServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        RequestDispatcher dis=null;
        BBSDAO dao=new BBSDAO();
        switch(request.getServletPath()) {
            case "/bbs/list":
                int page = request.getParameter("page")==null? 1:Integer.parseInt(request.getParameter("page"));
                int total = dao.total(); //전체페이지 계산
                int totalPage=total%5==0? total/5: (total/5)+1;
                int start = (page-1)*5;
                request.setAttribute("list", dao.list(start));
                request.setAttribute("page", page);
                request.setAttribute("totalPage", totalPage);
                dis=request.getRequestDispatcher("list.jsp");
                dis.forward(request, response);
                break;
            case "/bbs/insert":
                dis=request.getRequestDispatcher("insert.jsp");
                dis.forward(request, response);
                break;
            case "/bbs/read":
                String strSeqno=request.getParameter("seqno");
                int seqno=Integer.parseInt(strSeqno);
                request.setAttribute("vo", dao.read(seqno));
                dis=request.getRequestDispatcher("read.jsp");
                dis.forward(request, response);
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        BBSDAO dao=new BBSDAO();
        BBSVO vo=new BBSVO();
        vo.setTitle(request.getParameter("title"));
        vo.setContent(request.getParameter("content"));
        vo.setWriter(request.getParameter("writer"));
        switch(request.getServletPath()) {
            case "/bbs/insert":
                dao.insert(vo);
                break;
        }
        response.sendRedirect("list");
    }
}

```

---

## [게시판목록]

페이지:2/3

순번	제목	작성자	작성일	작성시간
7	건강을위해 밤에 야식을 삼가합니다.	guest	2020-05-17	12:00:00
6	게시판 개선을 축하합니다.	guest	2020-05-17	12:00:00
5	파일을 두개 첨부합니다.	guest	2020-05-17	12:00:00
4	3등 입장	spider	2020-05-17	12:00:00
3	2등 입장	iceman	2020-05-17	12:00:00

[이전] [다음]

[webContent]-[bbs]-[list.jsp]

```
<h2>[게시판목록]</h2>
<div>페이지:${page}/${totalPage}</div>
<table border=1 width=700>
  <tr><td>순번</td><td>제목</td><td>작성자</td><td>작성일</td><td>작성시간</td></tr>
  <c:forEach items="${list}" var="vo">
    <tr class="row">
      <td>${vo.seqno}</td>
      <td><a href="read?seqno=${vo.seqno}">${vo.title}</a></td>
      <td>${vo.writer}</td>
      <td><fmt:formatDate value="${vo.wdate}" pattern="yyyy-MM-dd"/></td>
      <td><fmt:formatDate value="${vo.wdate}" pattern="hh:mm:ss"/></td>
    </tr>
  </c:forEach>
</table>
<c:if test="${page==totalPage}"><a href="list?page=${page-1}">[이전]</a></c:if>
<c:if test="${page==1}"><a href="list?page=${page+1}">[다음]</a>
</c:if>
<c:if test="${page>1 && page<totalPage}">
  <a href="list?page=${page-1}">[이전]</a><a href="list?page=${page+1}">[다음]</a>
</c:if>
```

## [게시판 글쓰기]

제목	<input type="text"/>
<div></div>	
<div>저장 취소</div>	

## [게시판 글읽기]

[제목] 3등 입장  
[작성일시] 2020-05-16  
-----  
3등으로 입장이었습니다.

[webContent]-[bbs]-[insert.jsp]

```
<h2>[게시판 글쓰기]</h2>
<form action="insert" method="post">
  <input type="hidden" name="writer" value="guest">
  <table border=1 width=500>
    <tr><td width=100>제목</td><td><input type="text" name="title" size=54</td></tr>
    <tr><td colspan=2><textarea rows="10" cols="70" name="content"></textarea></td></tr>
  </table>
  <div id="btnMenu">
    <input type="submit" value="저장"> <input type="reset" value="취소">
  </div>
</form>
```

[webContent]-[bbs]-[read.jsp]

```
<h2>[게시판 글읽기]</h2>
<div>[제목] ${vo.title}</div>
<div>[작성일시] ${vo.wdate}</div>
-----
<div>${vo.content}</div>
```



## 11.5 도서목록 검색기능과 페이지징 기능

코드  검색 검색수: 24

코드	제목	저자	가격
1001	뇌를 자극하는 java 프로그래밍	김윤명	27,000원
1002	좋은 사자를 만드는 노출	정승익	20,000원
1003	예제로 배우는 엑셀 2007	김대식	19,000원
1004	지금 당장 경제공부 시작하라	최진기	18,500원
1005	Introduction to Algorithms	토마스	42,000원

◀ ▶ [1/5]

[Java Resources]-[src]-[book]-[BookDAO.java] 수정

//데이터갯수 출력

```
public int count(String key, String word) {
    int count=0;
    try {
        String sql ="select count(*) total from goodsinfo where " + key + " like ?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, "%" + word + "%");
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            count=rs.getInt("total");
        }
    } catch(Exception e) { System.out.println("데이터갯수 출력" + e.toString()); }
    return count;
}
```

//도서목록 출력

```
public ArrayList<BookVO> list(String key, String word, int page){
    ArrayList<BookVO> array=new ArrayList<BookVO>();
    try {
        String sql="select * from goodsinfo where " + key + " like ? limit ?, 5";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, "%" + word + "%");
        ps.setInt(2, (page-1)*5);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            BookVO vo=new BookVO();
            vo.setCode(rs.getString("code"));
            vo.setTitle(rs.getString("title"));
            vo.setWriter(rs.getString("writer"));
            vo.setPrice(rs.getInt("price"));
            array.add(vo);
        }
    } catch(Exception e) { System.out.println("도서목록:" + e.toString()); }
```

[Java Resources]-[src]-[book]-[BookServlet.java] 수정

```
,
switch(request.getServletPath()) {
    case "/book/list":
        String key=request.getParameter("key")==null?"code":request.getParameter("key");
        String word=request.getParameter("word")==null?"":request.getParameter("word");
        String strPage=request.getParameter("page")==null?"1":request.getParameter("page");

        int page=Integer.parseInt(strPage);
        int count = dao.count(key, word);
        int totPage=count%5==0?count/5:(count/5)+1;

        request.setAttribute("count", count);
        request.setAttribute("totPage", totPage);
        request.setAttribute("page", page);
        request.setAttribute("key", key);
        request.setAttribute("word", word);
        request.setAttribute("list", dao.list(key, word, page));

        dis=request.getRequestDispatcher("list.jsp");
        dis.forward(request, response);
        break;
}
```

```

<h2>[도서목록]</h2>
<div id="search">
  <form name="frm">
    <select name="key">
      <option value="code" <c:out value="\${key=='code'? 'selected':''}"/>>
        코드
      </option>
      <option value="title" <c:out value="\${key=='title'? 'selected':''}"/>>
        책제목
      </option>
      <option value="writer" <c:out value="\${key=='writer'? 'selected':''}"/>>
        책저자
      </option>
    </select>
    <input type="text" name="word" value="\${word}"/>
    <input type="submit" value="검색" id="btnSearch"/>
    검색수: <%out.println(request.getAttribute("count"));%>
    <input type="hidden" name="page" value="\${page}"/>
  </form>
</div>
...
<script>
  var totPage="\${totPage}";
  var page="\${page}";
  var count="\${count}";

  if(count==0){
    $("#pagination").hide();
  }

  if(page==1){
    $("#btnPre").attr("disabled", true);
  }

  if(page==totPage){
    $("#btnNext").attr('disabled', true);
  }

  $("#btnPre").on("click",function(){
    page--;
    $(frm.page).val(page);
    $(frm).submit();
  });

  $("#btnNext").on("click",function(){
    page++;
    $(frm.page).val(page);
    $(frm).submit();
  });

  $("#btnSearch").on("click", function(){
    $(frm.page).val(1);
    $(frm).submit();
  })
</script>

```

## 11.6 로그인 로그아웃, 회원가입 프로그램

```

public class UserVO {
  private String id;
  private String password;
  private String name;
  //...setter()/getter()/toString() 추가
}

```

```
public class UserDAO {
    public UserVO read(String id, String password) {
        UserVO vo=new UserVO();
        try {
            String sql="select * from userinfo where id=? and password=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, id);
            ps.setString(2, password);
            ResultSet rs=ps.executeQuery();
            if(rs.next()) {
                vo.setId(rs.getString("id"));
                vo.setName(rs.getString("name"));
            }
        }catch(Exception e) {
            System.out.println(e.toString());
        }
        return vo;
    }
    public void insert(UserVO vo) {
        try{
            String sql="insert into userinfo(id, password, name) values(?, ?, ?)";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getId());
            ps.setString(2, vo.getPassword());
            ps.setString(3, vo.getName());
            ps.execute();
        }catch(Exception e) {
            System.out.println(e.toString());
        }
    }
}
```

```
@WebServlet(value= {"/user/login", "/user/logout"})
public class UserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/user/logout":
                HttpSession session=request.getSession();
                session.invalidate();
                response.sendRedirect("/index.jsp");
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        UserDAO dao=new UserDAO();
        UserVO vo=new UserVO();

        switch(request.getServletPath()) {
            case "/user/login":
                String id=request.getParameter("id");
                String password=request.getParameter("password");

                vo=dao.read(id, password);
                JSONObject jObject=new JSONObject();
                jObject.put("id", vo.getId());
                out.println(jObject);

                HttpSession session=request.getSession();
                session.setAttribute("id", vo.getId());
                session.setAttribute("name", vo.getName());
                break;
        }
    }
}
```

# 한빛 미디어

아이디 : <input type="text" value="spider"/> 비밀번호 : <input type="password" value="...."/> <input type="button" value="로그인"/> <input type="button" value="회원가입"/>	<b>[회사소개]</b> 한빛미디어(주)는 가 실무경험과 현장 노	안녕하세요! 김재영님 <input type="button" value="로그아웃"/>	!통신 분야의 성장과 더불어 IT 전문가들의 풍부한 ! 분야의 IT 개발자들과 함께 발전해 왔습니다.
회사소개 책 정보 게시판 글쓰기 게시판 글읽기	<b>로그인성공시</b>	회사소개 책 정보 게시판 글쓰기 게시판 글읽기	

Copyright@ 1993-2020 한빛미디어(주)

## [webContent]-[user]-[login.jsp]

```
<script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
<form name="frm" style="margin:0px;">
  <div>아이디 : <input type="text" name="id" size=10></div>
  <div>비밀번호 : <input type="password" name="password" size=10></div>
  <div><input type="submit" value="로그인"><a href="/user/insert" style="font-size:12px;">회원가입</a></div>
</form>
<script>
  $(frm).submit(function(e){
    e.preventDefault();
    var id=$(frm.id).val();
    var password=$(frm.password).val();
    $.ajax({
      type:"post",
      url:"/user/login",
      data:{'id':id, 'password':password},
      dataType:"json",
      success:function(data){
        if(data.id==null){ alert("아이디 비밀번호가 일치하지 않습니다.");
        }else{ location.href="/index.jsp"; }
      }
    });
  });
</script>
```

## [webContent]-[memu.jsp] 수정

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<c:if test="${id==null}">
  <div class="login"><jsp:include page="/user/login.jsp"/></div>
</c:if>
<c:if test="${id!=null}">
  <div class="logout">
    <div>안녕하세요! ${name}님</div>
    <button onClick="location.href='/user/logout'">로그아웃</button>
  </div>
</c:if>
```

## [webContent]-[home.css] 수정

```
#menu .login, #menu .logout{ width:180px; border:1px dotted green; margin:10px 0px 20px 0px; padding:5px; }
```

## [webContent]-[bbs]-[insert.jsp] 수정

```
<form action="insert" method="post">
  <c:if test="${id==null}">
    <input type="hidden" name="writer" value="guest">
  </c:if>
  <c:if test="${id!=null}">
    <input type="hidden" name="writer" value="${id}">
  </c:if>
</form>
```

```

@WebServlet(value= {"/user/login", "/user/logout", "/user/insert"})
public class UserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/user/insert":
                response.sendRedirect("insert.jsp");
                break;
            ...
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        UserDao dao=new UserDao();
        UserVO vo=new UserVO();

        switch(request.getServletPath()) {
            case "/user/login":
                String id=request.getParameter("id");
                String password=request.getParameter("password");
                vo=dao.read(id, password);
                JSONObject jObject=new JSONObject();
                jObject.put("id", vo.getId());
                out.println(jObject);
                HttpSession session=request.getSession();
                session.setAttribute("id", vo.getId());
                session.setAttribute("name", vo.getName());
                break;
            case "/user/insert":
                vo.setId(request.getParameter("id"));
                vo.setPassword(request.getParameter("password"));
                vo.setName(request.getParameter("name"));
                dao.insert(vo);
                response.sendRedirect("/index.jsp");
                break;
        }
    }
}

```

## [회원가입]

아이디	<input type="text" value="spider"/>
비밀번호	<input type="password" value="****"/>
회원성명	<input type="text"/>

```

<h2>[회원가입]</h2>
<form action="insert" method="post">
    <table border=1 width=500>
        <tr>
            <td width=100>아이디 </td><td><input type="text" name="id" size=10></td>
        </tr>
        <tr>
            <td width=100>비밀번호</td><td><input type="password" name="password" size=10></td>
        </tr>
        <tr>
            <td width=100>회원성명</td><td><input type="text" name="name" size=10></td>
        </tr>
    </table>
    <div id="btnMenu">
        <input type="submit" value="저장"><input type="reset" value="취소">
    </div>
</form>

```

## 11.7 장바구니 관리 프로그램

[Java Resources]-[src]-[book]-[CartVO.java]

```
public class CartVO extends BookVO{
    private int number;
    //... setter()/getter()/toString() 추가
}
```

[Java Resources]-[src]-[book]-[BookServlet.java]

```
@WebServlet(value= {"/book/list", "/book/insert", "/book/delete", "/book/update", "/book/read", "/book/cart" })
public class BookServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        RequestDispatcher dis=null;
        BookDAO dao=new BookDAO();

        switch(request.getServletPath()){
            case "/book/cart":
                dis=request.getRequestDispatcher("listCart.jsp");
                dis.forward(request, response);
                break;
            ...
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
        request.setCharacterEncoding("UTF-8");
        BookDAO dao=new BookDAO();
        BookVO vo=new BookVO();

        switch(request.getServletPath()){
            case "/book/cart":
                String code=request.getParameter("code");
                vo=dao.read(code);
                CartVO cartVO=new CartVO();
                cartVO.setCode(vo.getCode());
                cartVO.setTitle(vo.getTitle());
                cartVO.setPrice(vo.getPrice());
                cartVO.setWriter(vo.getWriter());
                cartVO.setNumber(1);

                HttpSession session=request.getSession();
                ArrayList<CartVO> listCart=(ArrayList<CartVO>)session.getAttribute("listCart");

                //세션에 listCart가 존재하는 경우
                if(listCart!=null) {
                    boolean find=false;
                    for(CartVO cart:listCart) {
                        //listCart에서 code값이 일치하는 경우 수량을 1증가
                        if(cart.getCode().equals(code)) {
                            cart.setNumber(cart.getNumber() + 1);
                            find=true;
                        }
                    }
                    //listCart에 code값이 없는 경우
                    if(find==false) {
                        listCart.add(cartVO);
                    }
                }
                //세션에 listCart가 존재하지 않는 경우
            else {
                listCart=new ArrayList<CartVO>();
                listCart.add(cartVO);
            }
            session.setAttribute("listCart", listCart);
            break;
            ...
        }
        response.sendRedirect("list");
    }
}
```

## [도서목록]

코드	제목	저자	가격	장바구니담기
1002	좋은 사지을 만드는 노출	정승익	20,000	장바구니담기
1003	예제로 배우는 엑셀 2007	김대식	19,000	장바구니담기
1004	지금 당장 경제공부 시작하라!	최진기	19,500	장바구니담기
1005	Introduction to Algorithms	토마스 킴	42,000	장바구니담기
1006	자바프로그래밍	홍길동	100	장바구니담기

[webContent]-[book]-[list.jsp]

```

<h2>[도서목록]</h2>
<div class="item"><a href="/book/insert">도서등록</a></div>
<table id="tbl" border=1 width=700>
  <tr>
    <td width=100>코드</td><td width=200>제목</td><td width=100>저자</td><td width=100>가격</td><td>장바구니담기</td>
  </tr>
  <c:forEach items="${list}" var="vo">
    <tr class="row">
      <td><a href="read?code=${vo.code}">${vo.code}</a></td>
      <td>${vo.title}</td>
      <td>${vo.writer}</td>
      <td><fmt:formatNumber value="${vo.price}" pattern="#,###"/></td>
      <td><button code="${vo.code}">장바구니담기</button></td>
    </tr>
  </c:forEach>
</table>
...
<script>
  $("#tbl").on("click", ".row button", function(){
    var code=$(this).attr("code");
    if(!confirm(code + "을(를) 장바구니에 담겠습니까?")) return;
    $.ajax({
      type:"post",
      url:"/book/cart",
      data:{"code":code},
      success:function(){ alert("장바구니에 " + code + "을(를) 1개 담았습니다."); }
    });
  });
</script>

```

## [장바구니]

코드	제목	단가	수량	금액
1002	좋은 사지을 만드는 노출	20,000	2	40,000
1003	예제로 배우는 엑셀 2007	19,000	2	38,000
1004	지금 당장 경제공부 시작하라!	19,500	1	19,500

[webContent]-[book]-[listCart.jsp]

```

<h2>[장바구니]</h2>
<table id="tbl" border=1 width=700>
  <tr><td>코드</td><td>제목</td><td>단가</td><td>수량</td><td>금액</td></tr>
  <c:forEach items="${listCart}" var="vo">
    <tr class="row">
      <td><a href="read?code=${vo.code}">${vo.code}</a></td>
      <td>${vo.title}</td>
      <td><fmt:formatNumber value="${vo.price}" pattern="#,###"/></td>
      <td><fmt:formatNumber value="${vo.number}" pattern="#,###"/></td>
      <td><fmt:formatNumber value="${vo.price * vo.number}" pattern="#,###"/></td>
    </tr>
  </c:forEach>
</table>

```

## 11.8 MySql 저장프로시저

```
call out_total('김윤명', @o_total); select @o_total;
```

```
DROP PROCEDURE IF EXISTS OUT_TOTAL;
DELIMITER //
CREATE PROCEDURE OUT_TOTAL (in i_writer varchar(20), out o_total int)
BEGIN
    select sum(price) into o_total from goodsinfo where writer=i_writer;
END //
DELIMITER ;
```

```
call out_book_bbs();
```

```
DROP PROCEDURE IF EXISTS OUT_TOTAL;
DELIMITER //
CREATE PROCEDURE `OUT_BOOK_BBS` ()
BEGIN
    select * from goodsinfo;
    select * from bbs;
END //
DELIMITER ;
```

```
[Java Resources]-[src]-[book]-[BookDAO.java]
```

```
public HashMap<String, Object> call(){
    HashMap<String, Object> hMap=new HashMap<String, Object>();
    try {
        String sql="call out_book()";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.execute();

        ArrayList<HashMap<String, Object>> listBook=new ArrayList<HashMap<String, Object>>();
        ResultSet rs1=cs.getResultSet();
        while(rs1.next()) {
            HashMap<String, Object> bookMap=new HashMap<String, Object>();
            bookMap.put("title", rs1.getString("title"));
            listBook.add(bookMap);
        }
        hMap.put("books", listBook);

        //다음 ResultSet으로 이동
        cs.getMoreResults();

        ArrayList<HashMap<String, Object>> listBBS=new ArrayList<HashMap<String, Object>>();
        ResultSet rs2=cs.getResultSet();
        while(rs2.next()) {
            HashMap<String, Object> mapBBS=new HashMap<String, Object>();
            mapBBS.put("writer", rs2.getString("writer"));
            listBBS.add(mapBBS);
        }
        hMap.put("bbs", listBBS);
    }catch(Exception e) {
        System.out.println(e.toString());
    }
    return hMap;
}
```

```
call list_bbs('title', '입장', 0, 5, @count); select @count;
```

```
CREATE DEFINER='web'@'localhost' PROCEDURE `LIST_BBS`(
    in i_key varchar(20),
    in i_word varchar(20),
    in i_start int,
    in i_perPage int,
    out o_count int)
BEGIN
    IF(i_key='title')THEN
        select * from bbs where title like concat('%',i_word,'%') order by seqno desc limit i_start, i_perPage;
        select count(*) into o_count from bbs where title like concat('%',i_word,'%');
    END IF;
    IF(i_key='writer')THEN
        select * from bbs where writer like concat('%',i_word,'%') order by seqno desc limit i_start, i_perPage;
        select count(*) into o_count from bbs where writer like concat('%',i_word,'%');
    END IF;
END
```



```

public JSONObject list(String key, String word, int page, int perPage, String order, String desc) {
    JSONObject jObject=new JSONObject();
    try {
        String sql="call list_bbs(?,?,?,?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, key);
        cs.setString(2, word);
        cs.setInt(3, (page-1)*perPage);
        cs.setInt(4, perPage);
        cs.registerOutParameter(5, java.sql.Types.INTEGER);
        cs.execute();

        ResultSet rs=cs.getResultSet();
        JSONArray jArray=new JSONArray();
        while(rs.next()) {
            ...
        }

        jObject.put("list", jArray); //검색 목록
        int count=cs.getInt(5);
        jObject.put("count", count);
    } catch (Exception e) {
        System.out.println("게시판 목록:" + e.toString());
    }
    return jObject;
}

```

## [게시판 목록]

작성일자 ▼	guest	5행씩출력 ▼	검색	검색수: 4	작성일 ▼	내림차순 ▼	정렬
No.	제목	작성자	작성일				
28	게시판 개설을 축하합니다.	guest	2020/05/21 04:16:10				
27	MySQL 저장프로시저	guest	2020/05/21 02:12:19				
26	게시판 개설을 축하합니다.	guest	2020/05/21 02:06:16				
25	코로나 대응방안	guest	2020/05/21 02:05:04				

◀ ▶ [1/1]

```
call list('bbs', 'title', '입장', 'seqno', 'desc', 0, 5);
```

```

CREATE DEFINER='web'@'localhost' PROCEDURE `LIST`(
    in i_table varchar(50),
    in i_key varchar(50),
    in i_word varchar(50),
    in i_order varchar(50),
    in i_desc varchar(5),
    in i_start int,
    in i_perPage int)
BEGIN
    SET @i_table=i_table;
    SET @i_key=i_key;
    SET @i_word=concat('%',i_word,'%');
    SET @i_order=i_order;
    SET @i_desc=i_desc;
    SET @i_start=i_start;
    SET @i_perPage=i_perPage;

    SET @strSQL=concat('select * from ',@i_table,' where ',@i_key,' like ? order by ',@i_order,' ',@i_desc,' limit ?, ?');
    PREPARE ps from @strSQL;
    EXECUTE ps using @i_word, @i_start, @i_perPage;

    SET @strSQL=concat('select count(*) total from ',@i_table,' where ',@i_key,' like ?');
    PREPARE ps from @strSQL;
    EXECUTE ps using @i_word;
END

```

[[Java Resources]-[src]-[book]-[BBSDAO.java]

```
public JSONObject list(String key, String word, int page, int perPage, String order, String desc) {
    JSONObject jObject=new JSONObject();
    try {
        String sql="call list('bbs',?,?,?,??.?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, key);
        cs.setString(2, word);
        cs.setString(3, order);
        cs.setString(4, desc);
        cs.setInt(5, (page-1)*perPage);
        cs.setInt(6, perPage);
        cs.execute();

        ResultSet rs=cs.getResultSet();
        JSONArray jArray=new JSONArray();
        while(rs.next()) {
            ...
        }
        jObject.put("list", jArray); //검색 목록

        cs.getMoreResults(); //다음 Result로 이동

        rs=cs.getResultSet();
        int count=0;
        if(rs.next()) {
            count=rs.getInt("total");
        }
    }catch(Exception e) {
        System.out.println("게시판 목록:" + e.toString());
    }
    return jObject;
}
```

call add\_book('1001','자바 프로그래밍', '홍길동', 25000, @count); select @count;

```
CREATE DEFINER='web'@'localhost' PROCEDURE `ADD_BOOK`(
    in i_code char(5),
    in i_title varchar(50),
    in i_writer varchar(20),
    in i_price int,
    out o_count int)
BEGIN
    select count(*) cnt into o_count from goodsinfo where code=i_code;
    IF(o_count=0)THEN
        insert into goodsinfo(code,title,writer,price) values(i_code, i_title, i_writer, i_price);
    END IF;
END
```

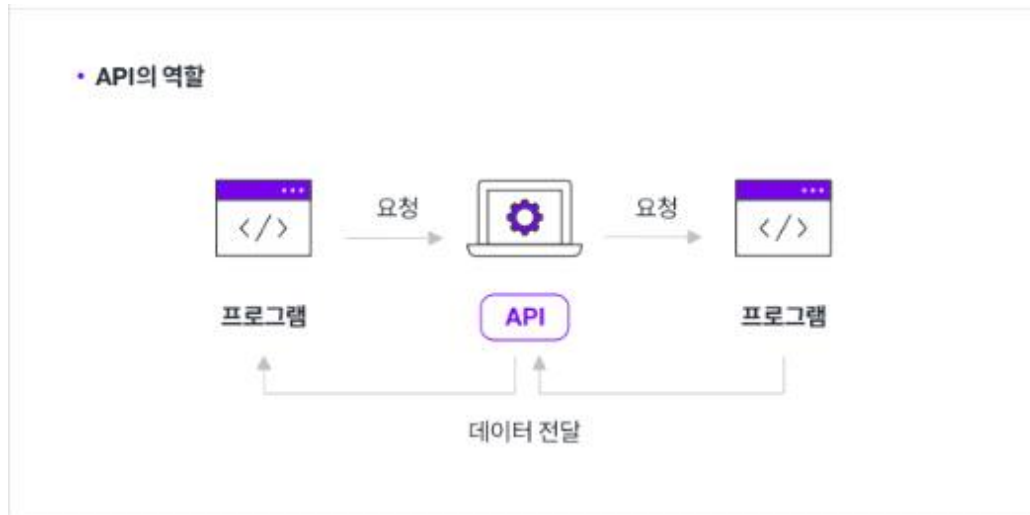
[[Java Resources]-[src]-[book]-[BookDAO.java]

```
public int insert(BookVO vo) {
    int count=0;
    try {
        String sql="call add_book(?,?,?,??.?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, vo.getCode());
        cs.setString(2, vo.getTitle());
        cs.setString(3, vo.getWriter());
        cs.setInt(4, vo.getPrice());
        cs.registerOutParameter(5, java.sql.Types.INTEGER);
        cs.execute();

        count=cs.getInt(5);
    }catch(Exception e) {
        System.out.println("도서등록:" + e.toString());
    }
    return count;
}
```

## 제12장 API 사용

API(Application Programming Interface)는 손님(프로그램)이 주문할 수 있게 점원이 메뉴(명령 목록)를 정리하고 주문(명령)을 받으면 요리사(응용프로그램)와 상호작용하여 요청된 메뉴(명령에 대한 값)를 전달한다. 쉽게 말해 API는 프로그램들이 서로 상호작용하는 것을 도와주는 매개체로 볼 수 있다.



### • API의 역할

1. API는 서버와 데이터베이스에 대한 출입구 역할을 한다.
2. API는 애플리케이션과 기기가 원활하게 통신할 수 있도록 한다.
3. API는 모든 접속을 표준화한다. :

### 12.1 네이버 애플리케이션 등록

- 1) <https://developers.naver.com>에 접속한다. :
- 2) 네이버 아이디로 접속한다.
- 3) [Application]-[내 애플리케이션 등록] 선택한다.
- 4) Client ID와 Client Secret이 발생한다.

### 12.2 네이버 블로그 검색 프로그램

블로그검색

도서검색

블로그 검색

검색데이터:699122

제목
자바학원 회사생활에도 큰 도움 됐어요
대구자바학원 걱정없이 해결해요!
엔제리너스 반미 샌드위치와 자바초코칩프라페
신촌자바학원 남들과 다르게 배워봐요!
자바학원 꼭 필요한 학습 끝!

[1/139825]

블로그검색

도서검색

블로그 검색

검색데이터:699122

제목
자바학원 회사생활에도 큰 도움 됐어요
대구자바학원 걱정없이 해결해요!
엔제리너스 반미 샌드위치와 자바초코칩프라페
신촌자바학원 남들과 다르게 배워봐요!
자바학원 꼭 필요한 학습 끝!

[1/139825]

```

public class NaverAPI {
    public static String main(String url, String query, int start, int display) {
        String clientId = "5kWNAPfRFDLK8SZVSkGP"; //애플리케이션 클라이언트 아이디값"
        String clientSecret = "8Auuy9tFIIf"; //애플리케이션 클라이언트 시크릿값"
        String text = null;
        try {
            text = URLEncoder.encode(query, "UTF-8");
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("검색어 인코딩 실패", e);
        }

        String apiURL = url+ "?query=" + text;
        apiURL += "&start="+start + "&display="+display;

        Map<String, String> requestHeaders = new HashMap<>();
        requestHeaders.put("X-Naver-Client-Id", clientId);
        requestHeaders.put("X-Naver-Client-Secret", clientSecret);
        String responseBody = get(apiURL,requestHeaders);

        System.out.println(responseBody);
        return responseBody;
    }

    private static String get(String apiUrl, Map<String, String> requestHeaders){
        HttpURLConnection con = connect(apiUrl);
        try {
            con.setRequestMethod("GET");
            for(Map.Entry<String, String> header :requestHeaders.entrySet()) {
                con.setRequestProperty(header.getKey(), header.getValue());
            }
            int responseCode = con.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK) { // 정상 호출
                return readBody(con.getInputStream());
            } else { // 예외 발생
                return readBody(con.getErrorStream());
            }
        } catch (IOException e) {
            throw new RuntimeException("API 요청과 응답 실패", e);
        } finally {
            con.disconnect();
        }
    }

    private static HttpURLConnection connect(String apiUrl){
        try {
            URL url = new URL(apiUrl);
            return (HttpURLConnection)url.openConnection();
        } catch (MalformedURLException e) {
            throw new RuntimeException("API URL이 잘못되었습니다. : " + apiUrl, e);
        } catch (IOException e) {
            throw new RuntimeException("연결이 실패했습니다. : " + apiUrl, e);
        }
    }

    private static String readBody(InputStream body) throws UnsupportedEncodingException{
        InputStreamReader streamReader = new InputStreamReader(body, "UTF-8");

        try (BufferedReader lineReader = new BufferedReader(streamReader)) {
            StringBuilder responseBody = new StringBuilder();
            String line;
            while ((line = lineReader.readLine()) != null) {
                responseBody.append(line);
            }
            return responseBody.toString();
        } catch (IOException e) {
            throw new RuntimeException("API 응답을 읽는데 실패했습니다.", e);
        }
    }
}

```

[JavaResources]-[src]-[util]-[NaverServlet.java]

```
@WebServlet(value={"/naver/blog","/naver/book"})
public class NaverServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();

        String url="";
        String query=request.getParameter("query");
        int start=(Integer.parseInt(request.getParameter("start")));
        int display=(Integer.parseInt(request.getParameter("display")));

        switch(request.getServletPath()) {
            case "/naver/blog":
                url="https://openapi.naver.com/v1/search/blog";
                break;
            case "/naver/book":
                url="https://openapi.naver.com/v1/search/book.json";
                break;
        }
        String result=NaverAPI.main(url,query,start,display);
        out.println(result);
    }
}
```

[WebContent]-[listBolog.jsp]

```
<html>
<head>
    <title>블로그검색</title>
    <meta name="viewport" content="user-scalable=no, width=device-width" charset="euc-kr"/>
    <link href="home_mobile.css" rel="stylesheet" type="text/css" media="screen and (max-width:799px)">
    <link href="home.css" rel="stylesheet" type="text/css" media="screen and (min-width:800px)">
    <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
</head>
<body>
<div id="divPage">
    <div id="divMenu"><jsp:include page="menu.jsp"/></div>
    <div id="divHeader"><h2>블로그 검색</h2></div>
    <div id="divCondition">
        <input type="text" id="query" value="자바">
        <input type="button" value="검색" id="btnSearch">
        <select id="display">
            <option value=5>5행 출력</option>
            <option value=10>10행 출력</option>
        </select>
        검색데이터:<span id="total"></span>
    </div>
    <table id="tbl" border=1></table>
    <script id="temp" type="text/x-handlebars-template">
        <tr class="title"><td width=500>제목</td></tr>
        {{#each items}}
        <tr class="row">
            <td><a href="{{link}}">{{title}}</a></td>
        </tr>
        {{/each}}
    </script>
    <div id="pagination">
        <button id="btnPre">◀</button>
        <button id="btnNext">▶</button>
        [<span id="curPage"></span></span><span id="totPage"></span>]
    </div>
</div>
</body>
</script>
    var url="/API/naver/blog";
</script>
<script src="home.js"></script>
</html>
```

#### [WebContent]-[home.js]

```
var page=1, display, query, total, totPage;
getList();

$("#btnSearch").on("click",function(){ page=1; getList(); });
$("#display, #query").on("change", function(){ page=1; getList();});
$("#btnNext").click(function(){ page++; getList(); });
$("#btnPre").click(function(){ page--; getList(); });

function getList(){
    query=$("#query").val();
    display=$("#display").val();
    var start=(page-1)*display + 1;
    $.ajax({
        type:"get",
        url:url,
        data:{"query":query,"display":display,"start":start},
        dataType:"json",
        success:function(data){
            var template = Handlebars.compile($("#temp").html());
            $("#tbl").html(template(data));

            total=data.total;
            if(total%display==0){ totPage=total/display;
            }else{ totPage=parseInt(total/display)+1; }

            if(page==1){
                $("#btnPre").attr("disabled",true);
            }else{ $("#btnPre").attr("disabled",false); }

            if(page==totPage){
                $("#btnNext").attr("disabled",true);
            }else{ $("#btnNext").attr("disabled",false);}

            $("#curPage").html(page);
            $("#totPage").html(totPage);
            $("#total").html(total);
        }
    });
}
```

#### [webContent]-[home.css]

```
@charset "UTF-8";
body{background:gray;}
#divPage{width:980px; background:white; box-shadow:10px 10px 10px blak; margin:auto; padding:0px 0px 20px 0px;}
#divMenu{margin:auto; overflow:hidden; background:#EAEAEA; padding:15px 0px 15px 20px;border-bottom:1px solid gray;}
#divMenu .item{width:100px; float:left; text-align:center;}
#divHeader{text-align:center;}
#divCondition{width:780px; margin:auto; overflow:hidden;padding:5px 10px 5px 10px; border:1px solid green;
    margin-bottom:10px; background:#EAEAEA;}
#divSearch{float:left; }
#divSort{float:right; }
#tbl{width:800px; margin:auto; overflow:hidden;}
#pagination{width:800px; margin:auto; text-align:center; padding:10px 0px 10px 0px;}
.price, .quantity, .sum {text-align:right;}
a{text-decoration:none; color:green;}
a:hover{font-weight:bold;}
select, input, button {height:23px;}
table {border-collapse:collapse;}
td{border:1px solid black; height:30px; font-size:12px; padding:0px 5px 0px 5px;}
table .title{background:#EAEAEA; text-align:center;}
table .row:hover{background:gray; color:white;}
input, button, select, textarea {font-size:12px;}
input[type='radio'], input[type='checkbox'] {height:12px;}
input[type='text']{height:20px;}
td button{text-align:center;}
textarea{margin:5px 0px 5px 0px;}
```

[webContent]-[home\_mobile.css]

```
#display{display:none;}
body{background:gray;}
#divPage{width:480px; background:white; box-shadow:10px 10px 10px blak; margin:auto; padding:0px 0px 20px 0px;}
#divMenu{margin:auto; overflow:hidden; background:#EAEAEA; padding:15px 0px 15px 20px;border-bottom:1px solid gray;}
#divMenu .item{width:100px; float:left; text-align:center;}
#divHeader{text-align:center;}
#divCondition{width:380px; margin:auto; overflow:hidden;padding:5px 10px 5px 10px; border:1px solid green;
...

```

[webContent]-[menu.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="item"><a href="/API/blogList.jsp">블로그검색</a></div>
<div class="item"><a href="/API/bookList.jsp">도서검색</a></div>

```

## 12.3 네이버 도서 검색 프로그램



[WebContent]-[listBook.jsp]

```
<div id="divPage">
  <div id="divMenu"><jsp:include page="menu.jsp"/></div>
  <div id="divHeader"><h2>도서 검색</h2></div>
  <div id="divCondition">
    <input type="text" id="query" value="자바">
    <input type="button" value="검색" id="btnSearch">
    <select id="display">
      <option value=4>4행 출력</option>
      <option value=8>8행 출력</option>
      <option value=12>12행 출력</option>
    </select>
    검색데이터:<span id="total"></span>
  </div>
  <div id="tbl"></div>
  <script id="temp" type="text/x-handlebars-template">
    {{#each items}}
      <div class="box"></div>
    {{/each}}
  </script>

```

## 12.4 다음(카카오) 장소 검색 및 지도 출력

블로그검색    도서검색    장소검색

### 장소 검색

인천 버거킹	검색	5할 출력 ▼	검색데이터:23
장소이름	주소	전화번호	위치보기
버거킹 인천송도DT점	인천 연수구 송도동 158-4	032-818-0332	위치보기
버거킹 인천송도센트럴파크점	인천 연수구 송도동 23-4	032-832-8531	위치보기
버거킹 삼산SK점	인천 부평구 삼산동 270-4	032-506-3643	위치보기
버거킹 인천작전점	인천 계양구 작전동 442-1	032-555-0332	위치보기
버거킹 인천구월점	인천 남동구 구월동 1456	070-7462-8793	위치보기

◀ ▶ [1/5]



#### DaumAPI.java

```

public class DaumAPI {
    public static String search(String apiURL) {
        try {
            URL url = new URL(apiURL);
            HttpURLConnection con = (HttpURLConnection)url.openConnection();
            con.setRequestMethod("GET");
            con.setRequestProperty("Authorization", "KakaoAK 4c9947b8b0448265ca15036c39a7fa4e");
            int responseCode = con.getResponseCode();
            BufferedReader br;

            // 정상 호출인 경우
            if(responseCode==200) {
                br = new BufferedReader(new InputStreamReader(con.getInputStream(), "UTF-8"));
            }
            //에러 발생가 발생한 경우
            else {
                br = new BufferedReader(new InputStreamReader(con.getErrorStream()));
            }

            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = br.readLine()) != null) {
                response.append(inputLine);
            }
            br.close();
            System.out.println(response.toString());
            return response.toString();
        } catch (Exception e) {
            return e.toString();
        }
    }
}

```



```

<html>
<head>
  <title>장소검색</title>
  <link rel="stylesheet" href="home.css">
  <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
  <script type="text/javascript"
    src="https://dapi.kakao.com/v2/maps/sdk.js?appkey=4c9947b8b0448265ca15036c39a7fa4e"></script>
  <style>
    #divMap { width:800px;
      height:230px;
      background:#EAEAEA;
      border:1px solid green;
      margin:auto;}
    #map{ width:800px; height:200px; }
  </style>
</head>
<body>
<div id="divPage">
  <div id="divMenu">
    <jsp:include page="menu.jsp"/>
  </div>
  <div id="divHeader"><h2>장소 검색</h2></div>
  <div id="divCondition">
    <input type="text" id="query" value="자바">
    <input type="button" value="검색" id="btnSearch">
    <select id="size">
      <option value=5>5행 출력</option>
      <option value=10>10행 출력</option>
    </select>
    검색데이터:<span id="total"></span>
  </div>
  <table id="tbl" border=1></table>
  <script id="temp" type="text/x-handlebars-template">
    <tr class="title">
      <td width=200>장소이름</td>
      <td width=300>주소</td>
      <td width=130>전화번호</td>
      <td width=70>위치보기</td>
    </tr>
    {{#each documents}}
    <tr class="row">
      <td>{{place_name}}</td>
      <td>{{address_name}}</td>
      <td>{{phone}}</td>
      <td><button x={{x}} y={{y}} place="{{place_name}}" tel="{{phone}}">위치보기</button></td>
    </tr>
    {{/each}}
  </script>

  <div id="pagination">
    <button id="btnPre">◀</button>
    <button id="btnNext">▶</button>
    [<span id="curPage"></span></span id="totPage"></span>]
  </div>

  <!-- 지도 출력 영역 -->
  <div id="divMap">
    <div id="map"></div>
    <div style="text-align:center;"><a id="btnClose" href="#">닫기</a></div>
  </div>
</div>
</body>

```

[내 애플리케이션]-[검색]-[플랫폼 설정하기]-[Web 플랫폼 등록]-[사이트 도메인]  
https://localhost:8088 등록

지도 appKey: JavaScript키를 사용  
검색 appKey: REST API키를 사용

```

<script>
    $("#divMap").hide();
    var page=1, size, query, total, totPage;
    getList();

    $("#tbl").on("click", ".row button", function(){
        $("#divMap").show();
        var lat=$(this).attr("y");
        var lng=$(this).attr("x");
        var place=$(this).attr("place");
        var tel=$(this).attr("tel");

        var container = document.getElementById('map');    //지도를 담을 영역의 DOM 레퍼런스
        //지도를 생성할 때 필요한 기본 옵션
        var options = { center: new kakao.maps.LatLng(lat, lng), level: 3 };
        var map = new kakao.maps.Map(container, options);    //지도 생성 및 객체 리턴
        var marker= new kakao.maps.Marker({ position: new kakao.maps.LatLng(lat, lng) });
        marker.setMap(map);

        var str = "<div style='padding:5px;font-size:12px;'>";
        str += place + "<br>" + tel;
        str += "</div>";
        var info=new kakao.maps.InfoWindow({ content:str });

        kakao.maps.event.addListener(marker, "mouseover", function(){ info.open(map, marker); });
        kakao.maps.event.addListener(marker, "mouseout", function(){ info.close(map, marker); });
    });

    $("#btnClose").on("click", function(){ $("#divMap").hide(); });
    $("#btnSearch").on("click",function(){ page=1; getList(); });
    $("#size, #query").on("change", function(){ page=1; getList(); });
    $("#btnNext").click(function(){ page++; getList(); });
    $("#btnPre").click(function(){ page--; getList(); });

    function getList(){
        query=$("#query").val();
        size=$("#size").val();
        $.ajax({
            type:"get",
            url:"https://dapi.kakao.com/v2/local/search/keyword.json",
            headers:{'Authorization':"KakaoAK 4c9947b8b0448265ca15036c39a7fa4e"},
            dataType:"json",
            data:{'query':query,"page":page,"size":size},
            success:function(data){
                var template = Handlebars.compile($("#temp").html());
                $("#tbl").html(template(data));
                total=data.meta.pageable_count;
                if(total % size==0){
                    totPage=total/size;
                }else{
                    totPage=parseInt(total/size)+1;
                }
                if(page==1){
                    $("#btnPre").attr("disabled",true);
                }else{ $("#btnPre").attr("disabled",false); }

                if(data.meta.is_end){
                    $("#btnNext").attr("disabled",true);
                }else{ $("#btnNext").attr("disabled",false);}

                $("#curPage").html(page);
                $("#totPage").html(totPage);
                $("#total").html(total);
            }
        });
    }
</script>
</html>

```

## 12.5 모바일웹(jQuery Mobile)

- <https://jquerymobile.com/download/> 접속하여 jQuery CDN를 복사한다.

book\_mobile.jsp

```
<html>
<head>
  <meta name="viewport" content="initial-scale=1, maximum-scale=1"/>
  <title>영화차트</title>
  <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" />
  <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
  <script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
</head>
<body>
  <div data-role="page" id="main">
    <div data-role="header" data-position="fixed" data-theme="e">
      <h1>도 서 검 색</h1>
    </div>
    <div>
      <input type="text" id="query" value="자바">
    </div>
    <div data-role="content" id="container">
      <ul id="book-list" data-role="listview"></ul>
      <script id="temp" type="text/x-handlebars-template">
        {{#each documents}}
        <li image="{{image}}" title="{{title}}">
          <a href="#">{{title}}</a>
        </li>
        {{/each}}
      </script>
    </div>
    <div>
      <input type="button" value="더보기" id="btnMore">
    </div>
    <div data-role="footer" data-theme="e"><h1>ICIA 교육원</h1></div>
  </div>
</body>
<script>
  var size=5, query;
  $("#btnMore").on("click", function(){
    size=size + 5;
    getList();
  });
  $("#query").keydown(function(key){
    if(key.keyCode==13){ size=5; getList(); }
  });
  getList();
  function getList(){
    query=$("#query").val();
    $.ajax({
      type:"get",
      url:"https://dapi.kakao.com/v3/search/book?target=title",
      headers:{"Authorization":"KakaoAK 4c9947b8b0448265ca15036c39a7fa4e"},
      dataType:"json",
      data:{"query":query,"size":size},
      success:function(data){
        var template = Handlebars.compile($("#temp").html());
        $("#book-list").html(template(data)).listview("refresh");
      }
    });
  }
  $(window).resize(function(){
    if(window.outerWidth > 799){
      window.location.href="book_pc.jsp";
    }else {
      window.location.href="book_mobile.jsp";
    }
  });
</script>
</html>
```



## 도서 검색



book\_pc.jsp

```
<style>
    .box{width:180px; height:250px; float:left;
    border:1px solid green;margin:5px;text-align:center}
    .box img{box-shadow:5px 5px 5px gray; margin:10px;}
</style>
</head>
<body>
<div id="divPage">
    <div id="divHeader"><h2>도서 검색</h2></div>
    <div id="divCondition">
        ...
    </div>
    <div id="tbl"></div>
    <script id="temp" type="text/x-handlebars-template">
    {{#each documents}}
        <div class="box"></div>
    {{/each}}
    </script>
    ...
</div>
</body>
<script>
    var page=1, size, query, total, totPage;
    getList();
    ...
    function getList(){
        query=$("#query").val();
        size=$("#size").val();
        $.ajax({
            type:"get",
            url:"https://dapi.kakao.com/v3/search/book?target=title",
            headers:{"Authorization":"KakaoAK 4c9947b8b0448265ca15036c39a7fa4e"},
            dataType:"json",
            data:{"query":query,"page":page,"size":size},
            success:function(data){
                var template = Handlebars.compile($("#temp").html());
                $("#tbl").html(template(data));
            }
        });
    }

    $(window).resize(function(){
        if(window.outerWidth > 799){
            window.location.href="book_pc.jsp";
        }else {
            window.location.href="book_mobile.jsp";
        }
    });
</script>
</html>
```

```
//이미지 파일 다운로드
String image="https://www.gstatic.com/webp/gallery/1.sm.jpg";
String file=image + image.substring(image.lastIndexOf("."));
//String file=System.currentTimeMillis() + ".gif";
InputStream in = null;
OutputStream out = null;
try{
    URL url = new URL(image);
    in=url.openStream();
    out=new FileOutputStream("c:/temp/" + file);

    while(true){
        int data = in.read();
        if(data == -1){
            break;
        }
        out.write(data);
    }
    in.close();
    out.close();
}catch(Exception e){
    System.out.println(e.toString());
}finally{
    if(in != null) in.close();
    if(out != null) out.close();
}
```

## 12.6 우편번호 검색

주소	<input type="text"/>	검색
	<input type="text" value="나머지주소입력"/>	



[WebContent]-[address.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
    <script src="https://code.jquery.com/jquery-2.2.1.js"></script>
    <script src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
    <title>주소입력</title>
    <style>
        table {border-collapse:collapse;}
        td {border:1px solid gray;
            height:30px;
            padding:5px;}
        .title{background:#EAEAEA;
            text-align:center;}
    </style>
</head>
<body>
    <form name="frm">
        <table>
            <tr>
                <td width=100 rowspan=2 class="title">주소</td>
                <td>
                    <input type="text" id="address1" size=30 readonly>
                    <input type="button" value="검색" id="btnSearch">
                </td>
            </tr>
            <tr>
                <td width=400>
                    <input type="text" id="address2" size=50 placeholder="나머지주소입력">
                </td>
            </tr>
        </table>
    </form>
</body>
<script>
    $("#btnSearch").click(function(){
        new daum.Postcode({
            oncomplete: function(data) {
                console.log(data)
                $("#address1").val(data.address); //data.roadAddress, data.jibunAddress, data.zonecode
            }
        }).open();
    });

    $("#address1").click(function(){
        $("#btnSearch").click();
    })
</script>
</html>
```

## 우편번호 찾기

☐ 지번주소
 ☒ 도로명주소

예:강남대로 100길

검색결과
인천 서구 비지니스로
인천 서구 비지니스로28번길
인천 서구 비지니스로181번길
인천 서구 비지니스로182번길
인천 서구 비지니스로193번길
인천 서구 비지니스로194번길

[WebContent]-[address1.jsp]

```

<script>
    $("#btnSearch").click(function(){
        window.open('search.jsp','search','width=350,height=400,top=200,left=100');
    });

    $(frm.address1).click(function(){
        $("#btnSearch").click();
    })
</script>

```

[WebContent]-[address1.jsp]

[illegible]

```

<table id="tbl"></table>
<script id="tempStreet" type="text/x-handlebars-template">
  <tr class="title">
    <td width=300>검색결과</td>
  </tr>
  {{#each documents}}
  <tr class="row">
    <td><a href="#">{{address.address_name}}</a></td>
  </tr>
  {{/each}}
</script>

<script id="tempRoad" type="text/x-handlebars-template">
  <tr><td class="title" width=300>검색결과</td></tr>
  {{#each documents}}
  <tr class="row">
    <td><a href="#">{{road_address.address_name}}</a></td>
  </tr>
  {{/each}}
</script>
</body>
<script>
  //radio 버튼을 클릭한 경우
  $("input[name='sel']").change(function(){
    $("#tbl").html("");
    $("#address").html("");
    if($(this).val()=='street'){
      $("#example").html("예:압구정동, 2글자 이상 입력");
    }else{
      $("#example").html("예:강남대로 100길");
    }
  });

  //검색버튼을 클릭한 경우
  $("#btnSearch").on("click", function(){
    getList();
  });

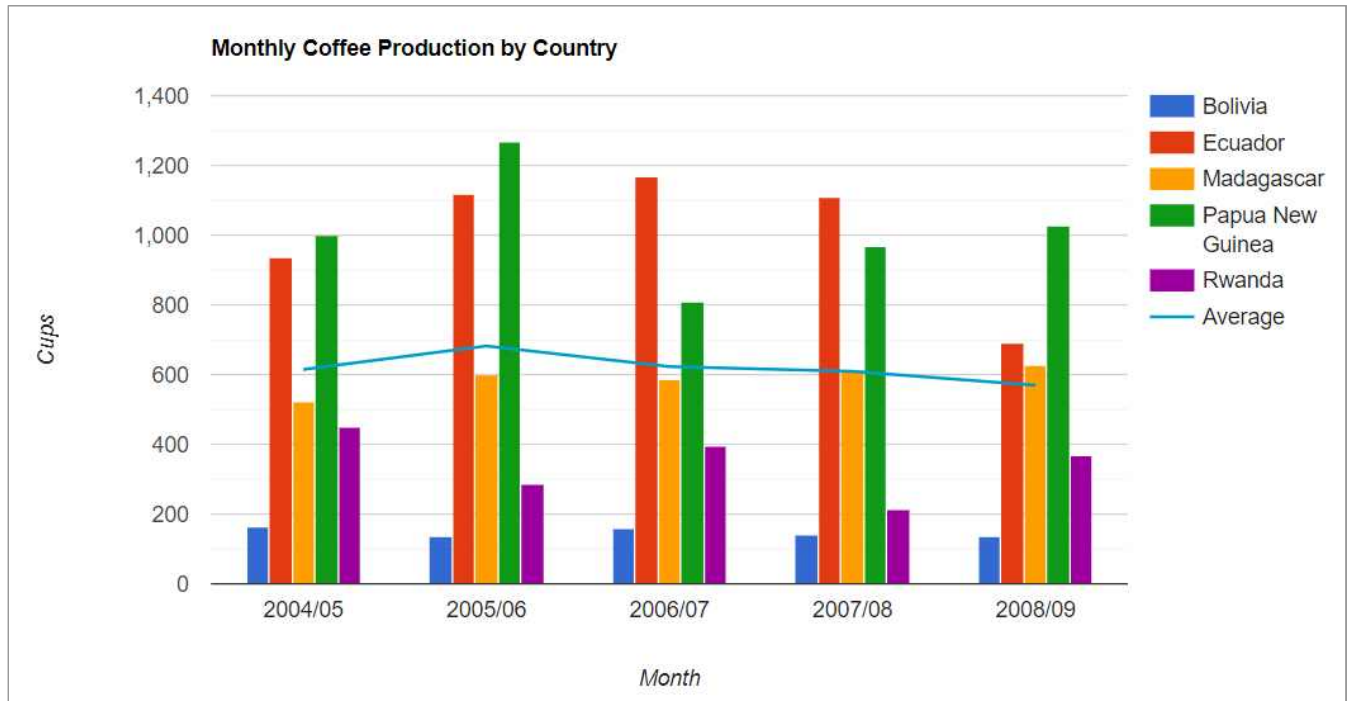
  $("#address").keydown(function(key){
    if(key.keyCode==13){
      getList();
    }
  });

  function getList(){
    query=$("#address").val();
    $.ajax({
      type:"get",
      url:"https://dapi.kakao.com/v2/local/search/address.json",
      headers:{"Authorization":"KakaoAK c9c00aae5ff46c1950267e681eac9d45"},
      dataType:"json",
      data:{"query":query},
      success:function(data){
        if($("#input[name='sel']:checked").val() == "street"){
          var template = Handlebars.compile($("#tempStreet").html());
        }else{
          var template = Handlebars.compile($("#tempRoad").html());
        }
        $("#tbl").html(template(data));
      }
    });
  }

  $("#tbl").on("click", ".row a", function(e){
    e.preventDefault();
    var address=$(this).html();
    $(opener.frm.address1).val(address);
    window.close();
  });
</script>
</html>

```

## 12.7 구글 차트 라이브러리



```
[src/main/java]-[com.example.web]-[HomeController.java]
```

```
<html>
<head>
  <title>구글 차트 라이브러리</title>
  <!-- 라이브러리 Load -->
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>

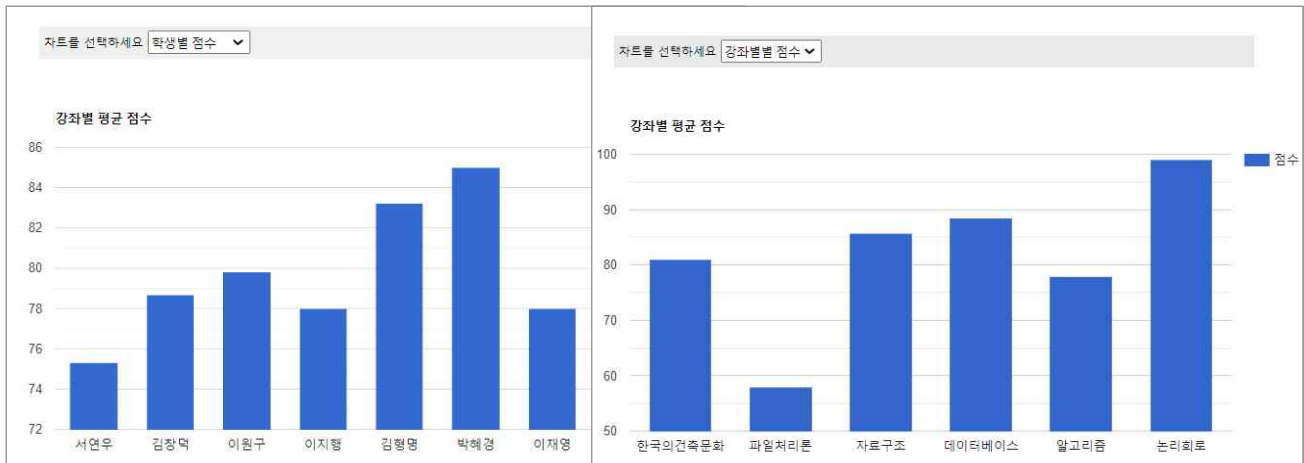
  <script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawVisualization);

    function drawVisualization() {
      /* 데이터 셋팅 */
      var data = google.visualization.arrayToDataTable([
        ['Month', 'Bolivia', 'Ecuador', 'Madagascar', 'Papua New Guinea', 'Rwanda', 'Average'],
        ['2004/05', 165, 938, 522, 998, 450, 614.6],
        ['2005/06', 135, 1120, 599, 1268, 288, 682],
        ['2006/07', 157, 1167, 587, 807, 397, 623],
        ['2007/08', 139, 1110, 615, 968, 215, 609.4],
        ['2008/09', 136, 691, 629, 1026, 366, 569.6] ]);

      /* 옵션 셋팅 */
      var options = {
        title : 'Monthly Coffee Production by Country',
        vAxis: {title: 'Cups'},
        hAxis: {title: 'Month'},
        seriesType: 'bars',
        series: {4: {type: 'line'}}
      };

      /* 차트 그리기 */
      var chart = new google.visualization.ComboChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="chart_div" style="width:900px; height: 500px;"></div>
</body>
</html>
```





[Java Resources]-[src]-[enrollments]-[EnrollmentsDAO.java]

//강좌별 성적

```
public JSONArray avgCourse() {
    JSONArray jArray=new JSONArray();
    try {
        String sql="select c.lcode, lname, avg(grade) average from courses c left join enrollments e on c.lcode=e.lcode
                    group by lcode, lname";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();

        JSONArray arr=new JSONArray();
        arr.add("과목명");
        arr.add("점수");
        jArray.add(arr);

        while(rs.next()) {
            arr=new JSONArray();
            arr.add(rs.getString("lname"));
            arr.add(rs.getDouble("average"));
            jArray.add(arr);
        }
    } catch (Exception e) {
        System.out.println("강좌별 평균점수:" + e.toString());
    }
    return jArray;
}
```

//학생별 성적

```
public JSONArray avgStudents() {
    JSONArray jArray=new JSONArray();
    try {
        String sql="select s.scode, sname, avg(grade) average from students s left join enrollments e on s.scode=e.scode
                    group by s.scode, sname";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();

        JSONArray arr=new JSONArray();
        arr.add("학생명");
        arr.add("점수");
        jArray.add(arr);

        while(rs.next()) {
            arr=new JSONArray();
            arr.add(rs.getString("sname"));
            arr.add(rs.getDouble("average"));
            jArray.add(arr);
        }
    } catch (Exception e) {
        System.out.println("학생별 평균점수:" + e.toString());
    }
    return jArray;
}
```

[Java Resources]-[src]-[enrollments]-[EnrollmentsServlet.java]

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();
    EnrollmentsDAO edao=new EnrollmentsDAO();

    switch(request.getServletPath()) {
    case "/enroll/avgStudents":
        out.println(edao.avgStudents());
        break;
    case "/enroll/avgCourses":
        out.println(edao.avgCourse());
        break;
    ...
    }
}
```

[webContent]-[chartEnroll.jsp]

```
<html>
<head>
    <title>구글 차트 라이브러리</title>
    <!-- 라이브러리 Load -->
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
    <script src="http://code.jquery.com/jquery-3.1.1.min.js"></script>
</head>
<body>
    <div style="height:25px;margin-left:150px;margin-top:100px;background:#EAEAEA;width:700px;padding:5px;">
        <span style="font-size:13px;">차트를 선택하세요 </span>
        <select id="sel" style="height:25px;">
            <option value="/haksa/enroll/avgStudents">학생별 점수</option>
            <option value="/haksa/enroll/avgCourses">강좌별 점수</option>
        </select>
    </div>
    <div id="chart_div" style="width:1000px; height:500px;"></div>
</body>
<script type="text/javascript">
    changeChart();
    $("#sel").on("change", function(){
        changeChart();
    });

    function changeChart(){
        var url=$("#sel").val();
        google.charts.load('current', {'packages':['corechart']});
        google.charts.setOnLoadCallback(drawVisualization);
        function drawVisualization() {
            $.ajax({
                type:"get",
                url:url,
                dataType:"json",
                success:function(result){
                    /* 데이터 셋팅 */
                    var data=google.visualization.arrayToDataTable(result);

                    /* 옵션 셋팅 */
                    var options = {
                        title : '강좌별 평균 점수',
                        seriesType: 'bars'
                    };

                    /* 차트 그리기 */
                    var chart = new google.visualization.ComboChart(document.getElementById('chart_div'));
                    chart.draw(data, options);
                }
            });
        }
    }
</script>
</html>
```

## 12.8 CK Editor

1) 입력양식을 작성한다.

[WebContent]-[ckeditor.html]

```
<html>
<head>
  <meta charset="UTF-8">
  <script src="https://cdn.ckeditor.com/4.16.2/standard/ckeditor.js"></script>
  <style>
    input[type=text]{border: 1px solid gray; height:23px; width:100%}
    form {width: 700px; margin:30px;}
  </style>
</head>
<body>
  <form name="frm" method="post" action="insert">
    <input type="text" name="title" placeholder="제목을 입력하세요."/>
    <hr/>
    <textarea id="editor" name="content" cols=80 rows=15></textarea>
    <hr/>
    <input type="submit" value="저장"/>
    <input type="reset" value="취소"/>
  </form>
</body>
<script>
  var ckeditor_config = {
    resize_enable : false, //editor 사이즈를 변경하지 못한다.
    enterMode : CKEDITOR.ENTER_BR,
    shiftEnterMode : CKEDITOR.ENTER_P,
    filebrowserUploadUrl : "/ckupload",
    height: 300
  };
  CKEDITOR.replace('editor', ckeditor_config)
</script>
</html>
```

- 2) 파일관리자에서 [c:]-[temp]-[ck] 폴더를 생성한다.  
3) server.xml에 아래 내용을 추가한다.

```
[Servers]-[server.xml]
```

```
<Context docBase="c:/temp" path="/upload" reloadable="true"/>
```

- 4) 파일 업로드를 위해 cos.jar 파일을 아래로 복사한다.

```
[WebContent]-[WEB-INF]-[lib]-[cos.jar]
```

- 5) 업로드 Controller를 작성한다.

```
[Java Resources]-[src]-[bbs]-[BBSServlet.java]
```

```
@WebServlet(value= {"/insert", "/ckupload"})
```

```
public class BBSServlet extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
        switch(request.getServletPath()){
```

```
            case "/insert":
```

```
                RequestDispatcher dis=request.getRequestDispatcher("ckeditor.html");
```

```
                dis.forward(request, response);
```

```
                break;
```

```
        }
```

```
    }
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
        request.setCharacterEncoding("UTF-8");
```

```
        response.setContentType("text/html;charset=UTF-8");
```

```
        PrintWriter out=response.getWriter();
```

```
        switch(request.getServletPath()){
```

```
            case "/insert":
```

```
                String title = request.getParameter("title");
```

```
                String content = request.getParameter("content");
```

```
                System.out.println("content..." + content);
```

```
                out.println("<div>" + content + "</div>");
```

```
                break;
```

```
            case "/ckupload":
```

```
                String path ="c:/temp/ck"; //업로드 될 실제 디렉터리
```

```
                MultipartRequest multi = new MultipartRequest(
```

```
                    request,
```

```
                    path,
```

```
                    1024 * 1024 * 10,
```

```
                    "UTF-8",
```

```
                    new DefaultFileRenamePolicy());
```

```
                String fileName=multi.getFileName("upload");
```

```
                String fileUrl="/upload/ck" + fileName;
```

```
                JSONObject json = new JSONObject();
```

```
                json.put("uploaded", 1);
```

```
                json.put("fileName", fileName);
```

```
                json.put("url", fileUrl);
```

```
                out.println(json);
```

```
            }
```

```
        }
```

```
    }
```