

8장. JSP 기본 동작

1. JSP (Java Server Pages)

- 서버릿은 성능좋은 웹 응용프로그램을 효과적으로 개발할 수는 있지만, 프로그램 작성이 불편한 단점이 있다.
- JSP는 서버릿 기술을 바탕으로 ASP처럼 쉽고, 빠르게 원하는 웹응용 프로그램을 작성할 수 있다.
- java in HTML

(1) Servlet과 JSP과 비교

- ① JSP는 Servlet보다 쉽고 빠르게 프로그램 작성
- ② JSP 화면구성을 위한 HTML부분과 프로그램 로직(Logic)을 분리할 수 있다
(프로그래머와 웹디자이너가 협력작업을 할 수 있다)
- ③ 서버릿은 자바 프로그램이기 때문에 HTML의 페이지 디자인을 하기 위한 디자인도구가 부족한데, JSP는 HTML페이지 디자인도구들을 이용해서 작성할 수 있다.
- ④ hello.jsp : jsp 파일
↓
hello_java.jsp : 내부에서 서버릿 작성된다.
↓
hello_java.class : 내용이 바뀔때만 자동 컴파일

2. JSP 기본문법

(1) 스크립트 요소 4가지

- ① 선언문
 <%!
 전역변수 및 메소드를 선언하는 영역
 %>
- ② 스크립트릿
 <%
 코드작성 (지역변수선언, for, while, if 등)
 %>
- ③ 표현식
 <%=변수 또는 메소드 %>
- ④ 주석
 - i) HTML에서 주석
 <!-- Fighting <%=name%> -->
 - ii) JSP에서 주석
 <%-- Fighting<%=name%> --%>
 - iii) 스크립트 언어주석
 /* 주석(여러줄 주석) */
 //주석(한줄 주석)

(2) JSP의 지시자 (Directive)

- 클라이언트가 요청한 JSP 페이지가 실행될 때, 필요한 정보를 지정하는 역할을 한다.
- 필요한 정보를 JSP 컨테이너에게 알려서 어떻게 처리하도록 하는 지시자이다.
- 지시자는 태그 안에서 @로 시작하며, page, include, taglib가 있다.

① page 지시자 <%@ page %>

- JSP페이지에 대한 속성 설정
서버에 요청한 JSP 페이지가 실행될 때, 생성되는 페이지의 타입, 스크립트 언어, import 할 클래스, 세션 및 버퍼의 사용 여부 및 버퍼의 크기 등의 JSP 페이지에서 필요한 설정 정보를 지정
- info : 페이지를 설명해 주는 문자열, 속성값의 내용이나 길이의 제한이 없음
(JSP페이지의 제목을 붙이는 것 같은 기능)
<%@ page info="JSPStudy.co.kr" %>
- language : 기본값 java(생략가능)
<%@ page language="java" %>
- contentType : 어떤 내용으로 출력할것인지 MIME형태 , ISO-8859-1(표준)
<%page contentType="text/html; charset=UTF-8">
- extends : 상속받은 클래스
<% page extends="com.jspstudy.Directive" %>
- import : package이름을 지정하지 않고 다른 클래스를 가져와서 사용
(쉽표로 구분해서 사용가능, 유일하게 중복사용가능)
<%@ page import="java.util.*, java.sql.*" %>
- session : HttpSession을 사용할지여부
<% page session="false" %>
- buffer : nono으로 지정하면 출력버퍼 사용안함
<%@ page buffer="16kb" %>
- autoFlush : true로 설정할경우 자동으로 버퍼삭제(다 채워졌을때)
<%@ page autoFlush="false" %>
- isThreadSafe : 동시에 여러브라우저의 요청을 처리할지 여부
<%@ page isThreadSafe="false" %>
- errorPage : 예외가 발생할 경우 넘겨주는 페이지
<%@ page errorPage="Error.jsp" %>
//Error페이지가 나타나지 않을 경우
Explorer에서 도구 -> 인터넷옵션 -> 고급 -> HTTP오류페이지표시(체크해제)
- isErrorPage : 예외처리 담당 페이지인지 아닌지 여부
<%@ page isErrorPage="true" %>
- pageEncoding : 인코딩 지정(기본값 : ISO-8859-1)
<%@ page pageEncoding="UTF-8" %>

② include 지시자: JSP파일 삽입

③ taglib

- 표현 언어(EL : Expression Language), JSTL (JSP Standard Tag Library),
커스텀 태그(Custom Tag)를 JSP 페이지 내에 사용할 때 쓴다.
- <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
uri : 태그의 설정 정보
prefix : uri 대신 사용되는 네임스페이스

(3) Post방식과 Get방식의 차이

① get

- 주로 웹브라우저가 웹서버에 데이터를 요청할때 사용
- 전송데이터 길이의 제한이 있다. (ex : 문자열전송)
- 데이터가 인코딩되어 URL에 붙는다.

② post

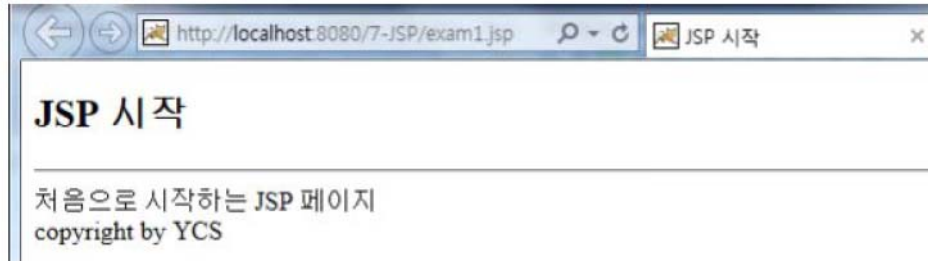
- 웹브라우저에서 웹서버에 데이터를 전달
- 전송데이터 길이의 제한이 없다. (ex : 대량의 데이터, 파일전송)
- 전달되는 데이터가 보이지 않는다.

예제1. JSP 시작하기

Dynamic Web Project : 8-JSP

JSP File : WebContent/exam1.jsp

<실행 결과>



예제2. 오늘 날짜 출력하기

Dynamic Web Project : 8-JSP

JSP File : WebContent/exam2.jsp

<실행 결과>

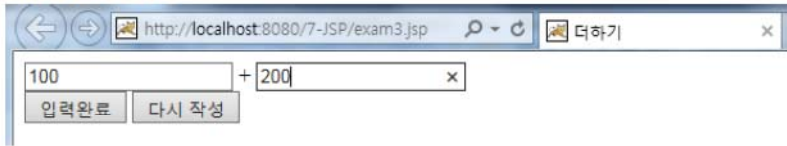


예제3. 두 수의 합 구하기

Dynamic Web Project : 8-JSP

JSP File : WebContent/exam3.jsp
WebContent/exam3Pro.jsp

<실행 결과>

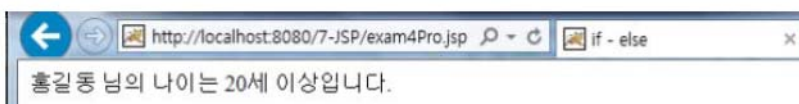


예제4. 입력받은 나이가 20보다 작은지 큰지 비교

Dynamic Web Project : 8-JSP

JSP File : WebContent/exam4.jsp
WebContent/exam4Pro.jsp

<실행 결과>

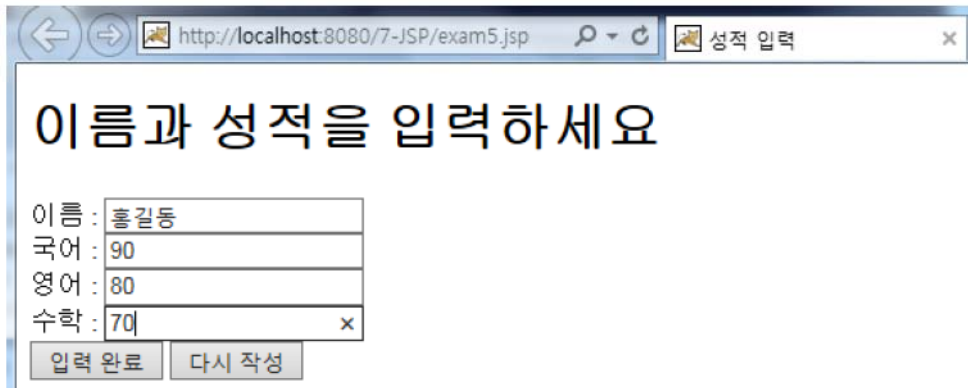


예제5. 성적 처리

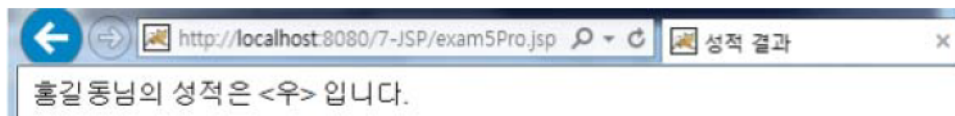
Dynamic Web Project : 8-JSP

JSP File : WebContent/exam5.jsp
WebContent/exam5Pro.jsp

<실행 결과>



A screenshot of a web browser window showing the URL `http://localhost:8080/7-JSP/exam5.jsp`. The page title is "성적 입력". The main heading is "이름과 성적을 입력하세요". Below the heading, there are four input fields: "이름:" with the value "홍길동", "국어:" with the value "90", "영어:" with the value "80", and "수학:" with the value "70". At the bottom of the form, there are two buttons: "입력 완료" and "다시 작성".



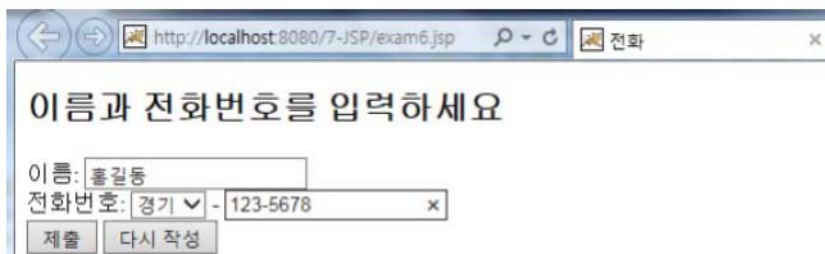
A screenshot of a web browser window showing the URL `http://localhost:8080/7-JSP/exam5Pro.jsp`. The page title is "성적 결과". The main text is "홍길동님의 성적은 <우> 입니다."

예제6. 전화 지역번호 확인하기

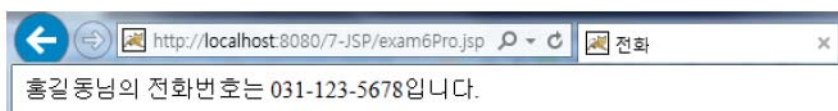
Dynamic Web Project : 8-JSP

JSP File : WebContent/exam6.jsp
WebContent/exam6Pro.jsp

<실행 결과>



A screenshot of a web browser window showing the URL `http://localhost:8080/7-JSP/exam6.jsp`. The page title is "전화". The main heading is "이름과 전화번호를 입력하세요". Below the heading, there are two input fields: "이름:" with the value "홍길동" and "전화번호:" with a dropdown menu showing "경기" and a text box with the value "123-5678". At the bottom of the form, there are two buttons: "제출" and "다시 작성".

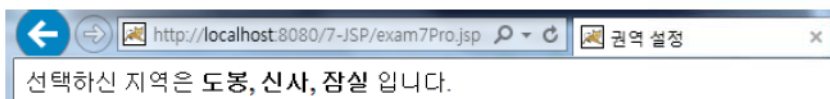
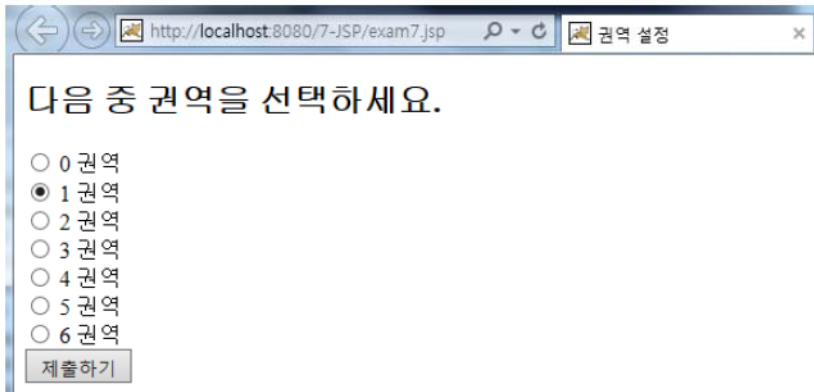


A screenshot of a web browser window showing the URL `http://localhost:8080/7-JSP/exam6Pro.jsp`. The page title is "전화". The main text is "홍길동님의 전화번호는 031-123-5678입니다."

예제7. 권역 선택하기

Dynamic Web Project : 8-JSP
JSP File : WebContent/exam7.jsp
WebContent/exam7Pro.jsp

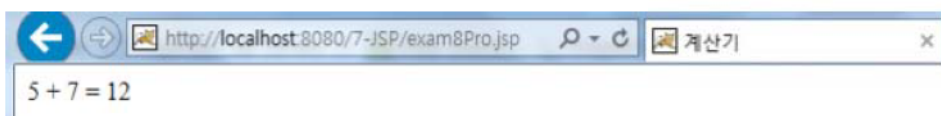
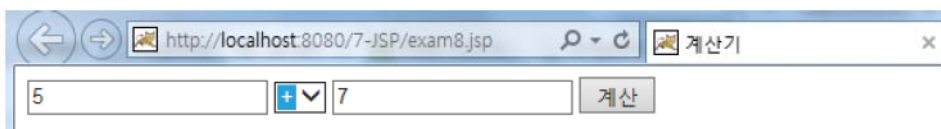
<실행 결과>



예제8. 계산기

Dynamic Web Project : 8-JSP
JSP File : WebContent/exam8.jsp
WebContent/exam8Pro.jsp

<실행 결과>



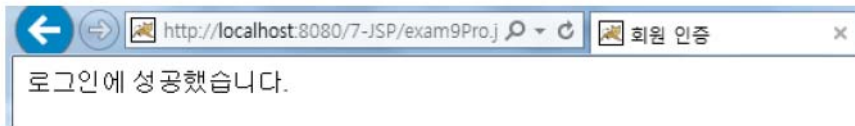
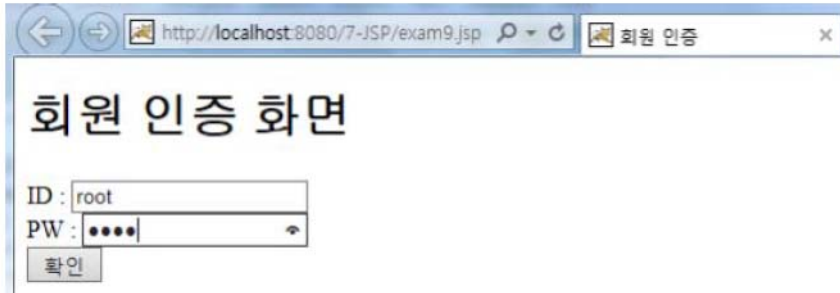
예제9. 회원 인증 확인하기

Dynamic Web Project : 8-JSP

JSP File : WebContent/exam9.jsp

WebContent/exam9Pro.jsp

<실행 결과>



문제1. 계산기

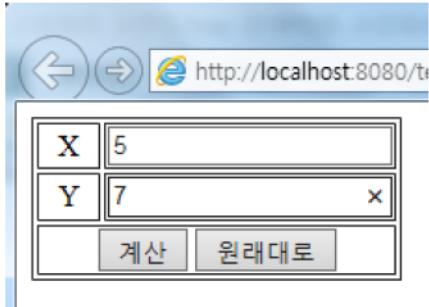
Dynamic Web Project : 8-JSP

Folder : exam

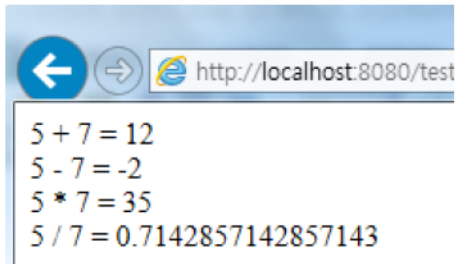
JSP File : WebContent/exam/input.jsp
WebContent/exam/result.jsp

<실행 결과>

- input.jsp



- result.jsp



5 + 7 = 12
5 - 7 = -2
5 * 7 = 35
5 / 7 = 0.7142857142857143

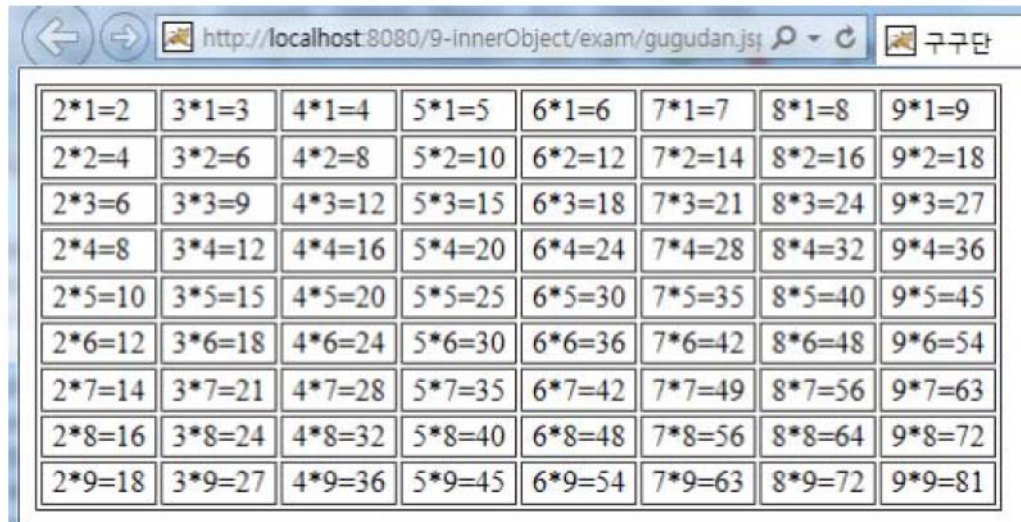
문제2. 구구단 만들기

Dynamic Web Project : 8-JSP

Folder : exam

JSP File : WebContent/exam/gugudan.jsp

<실행 결과>



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/9-innerObject/exam/gugudan.jsp`. The page content is a multiplication table (gugudan) with 9 rows and 9 columns. Each cell contains a multiplication problem and its result, such as `2*1=2` in the first row, first column, and `9*9=81` in the last row, last column.

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 2*1=2 | 3*1=3 | 4*1=4 | 5*1=5 | 6*1=6 | 7*1=7 | 8*1=8 | 9*1=9 |
| 2*2=4 | 3*2=6 | 4*2=8 | 5*2=10 | 6*2=12 | 7*2=14 | 8*2=16 | 9*2=18 |
| 2*3=6 | 3*3=9 | 4*3=12 | 5*3=15 | 6*3=18 | 7*3=21 | 8*3=24 | 9*3=27 |
| 2*4=8 | 3*4=12 | 4*4=16 | 5*4=20 | 6*4=24 | 7*4=28 | 8*4=32 | 9*4=36 |
| 2*5=10 | 3*5=15 | 4*5=20 | 5*5=25 | 6*5=30 | 7*5=35 | 8*5=40 | 9*5=45 |
| 2*6=12 | 3*6=18 | 4*6=24 | 5*6=30 | 6*6=36 | 7*6=42 | 8*6=48 | 9*6=54 |
| 2*7=14 | 3*7=21 | 4*7=28 | 5*7=35 | 6*7=42 | 7*7=49 | 8*7=56 | 9*7=63 |
| 2*8=16 | 3*8=24 | 4*8=32 | 5*8=40 | 6*8=48 | 7*8=56 | 8*8=64 | 9*8=72 |
| 2*9=18 | 3*9=27 | 4*9=36 | 5*9=45 | 6*9=54 | 7*9=63 | 8*9=72 | 9*9=81 |

9장. JSP 내장 객체

1. JSP 내장 객체

- 웹 컨테이너는 JSP 페이지에서 사용되는 9개의 객체를, 객체의 생성 없이 바로 사용할 수 있도록 제공한다. 이들 객체들을 JSP의 내장 객체(Implicit Object)라고 부른다.

- ① request (javax.servlet.http.HttpServletRequest)
 - 파라미터를 포함한 요청을 담고 있는 객체
- ② response (javax.servlet.http.HttpServletResponse)
 - 요청에 대한 응답
- ③ out (javax.servlet.jsp.JspWriter)
 - 페이지의 내용을 담고 있는 출력 스트림 객체
- ④ session (javax.servlet.http.HttpSession)
 - 세션 정보를 담고있는 객체
- ⑤ application (javax.servlet.ServletContext)
 - 모든 페이지가 공유할 데이터를 담고있는 객체
- ⑥ pageContext (javax.servlet.jsp.PageContext)
 - 페이지 실행에 필요한 context정보를 담고있는 객체
- ⑦ page (javax.servlet.jsp.HttpJspPage)
 - JSP 페이지의 서블릿 객체
- ⑧ config (javax.servlet.ServletConfig)
 - JSP 페이지의 서블릿 설정 데이터 초기화 정보 객체
- ⑨ exception (java.lang.Throwable)
 - JSP 페이지의 서블릿 실행시 처리하지 못한 예외처리 객체

2. request 객체

- 클라이언트가 전송한 요청 정보를 제공한다.

(1) 기능

- 클라이언트와 관련된 정보 읽기
- 서버와 관련된 정보 읽기
- 클라이언트가 전송한 요청 파라미터 읽기
- 클라이언트가 전송한 요청 헤더 읽기
- 클라이언트가 전송한 쿠키 읽기
- 속성 처리

(2) 클라이언트 및 서버 정보 관련 메소드

| 메소드 | 리턴 타입 | 설명 |
|------------------------|--------|--|
| getRemoteAddr() | String | 웹 서버에 연결한 클라이언트의 IP주소를 구한다. |
| getContentLength() | long | 클라이언트가 전송한 요청 정보의 길이를 구한다. 전송된 데이터의 길이를 알 수 없는 경우 -1을 리턴. |
| getCharacterEncoding() | String | 클라이언트가 요청 정보를 전송할 때 사용한 문자 인코딩을 구한다. |
| getContentType() | String | 클라이언트가 요청 정보를 전송할 때 사용한 콘텐츠의 타입을 구한다. |
| getProtocol() | String | 클라이언트가 요청한 프로토콜을 구한다. |
| getMethod() | String | 웹 브라우저가 정보를 전송할 때 사용한 방식을 구한다. (GET, POST) |
| getResultURI() | String | 웹브라우저가 요청한 URI를 구한다. |
| getResultURL() | String | 웹브라우저가 요청한 URL을 구한다. |
| getContextPath() | String | JSP 페이지가 속한 웹 어플리케이션의 컨텍스트 경로를 구한다. |
| getServerName() | String | 연결할 때 사용한 서버 이름을 구한다. |
| getServerPort() | int | 서버가 실행중인 포트 번호를 구한다. |

(3) 파라미터 읽기 메소드

| 메소드 | 리턴 타입 | 설명 |
|---------------------------------|---------------------------|---|
| getParameter(String name) | String | 이름이 name인 파라미터 값을 구한다. 존재하지 않으면 null 리턴 |
| getParameterValues(String name) | String[] | 이름이 name인 파라미터 값을 배열로 구한다. 존재하지 않으면 null 리턴 |
| getParameterNames() | java.util. Enumeration | 웹 브라우저가 전송한 파라미터의 이름을 구한다. |
| getParameterMap() | java.util.Map | 웹 브라우저가 전송한 파라미터의 맵을 구한다. 맵은 <이름, 값> 쌍으로 구성 |

(4) 헤더 읽기 메소드

| 메소드 | 리턴 타입 | 설명 |
|------------------------|---------------------------|--------------------|
| getHeader(String name) | String | 지정한 이름의 헤더 값을 구한다. |
| getHeaderNames() | java.util. Enumeration | 모든 헤더의 이름을 구한다. |

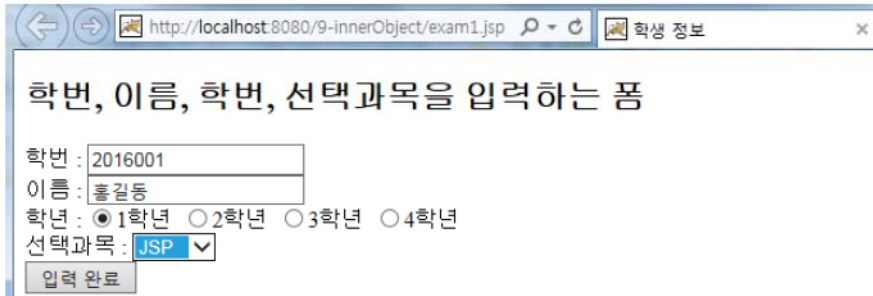
예제1. 학생정보 입력 및 확인하기

Dynamic Web Project : 9-innerObject

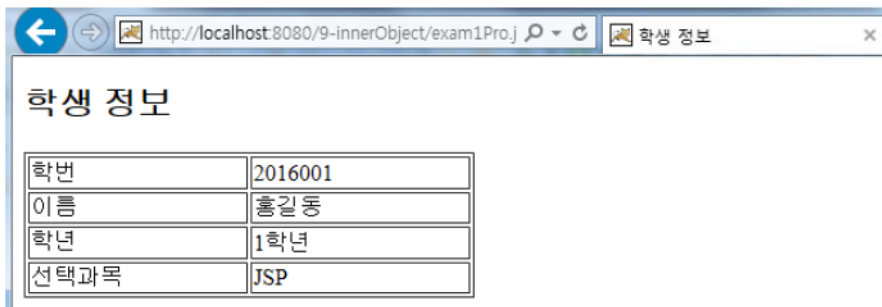
JSP File : WebContent/exam1.jsp

WebContent/exam1Pro.jsp

<실행 결과>



A screenshot of a web browser window showing the URL `http://localhost:8080/9-innerObject/exam1.jsp`. The page title is "학생 정보". The main heading is "학번, 이름, 학번, 선택과목을 입력하는 폼". Below the heading are four input fields: "학번:" with the value "2016001", "이름:" with the value "홍길동", "학년:" with radio buttons for "1학년" (selected), "2학년", "3학년", and "4학년", and "선택과목:" with a dropdown menu showing "JSP". At the bottom is a button labeled "입력 완료".



A screenshot of a web browser window showing the URL `http://localhost:8080/9-innerObject/exam1Pro.jsp`. The page title is "학생 정보". The main heading is "학생 정보". Below the heading is a table displaying the entered student information.

| | |
|------|---------|
| 학번 | 2016001 |
| 이름 | 홍길동 |
| 학년 | 1학년 |
| 선택과목 | JSP |

예제2. 계산기 예제에서 클라이언트 정보와 서버정보 읽기

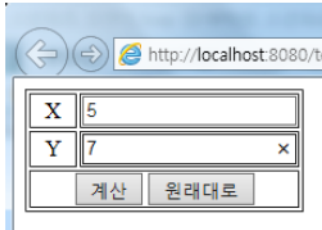
Dynamic Web Project : 9-innerObject

Folder : exam

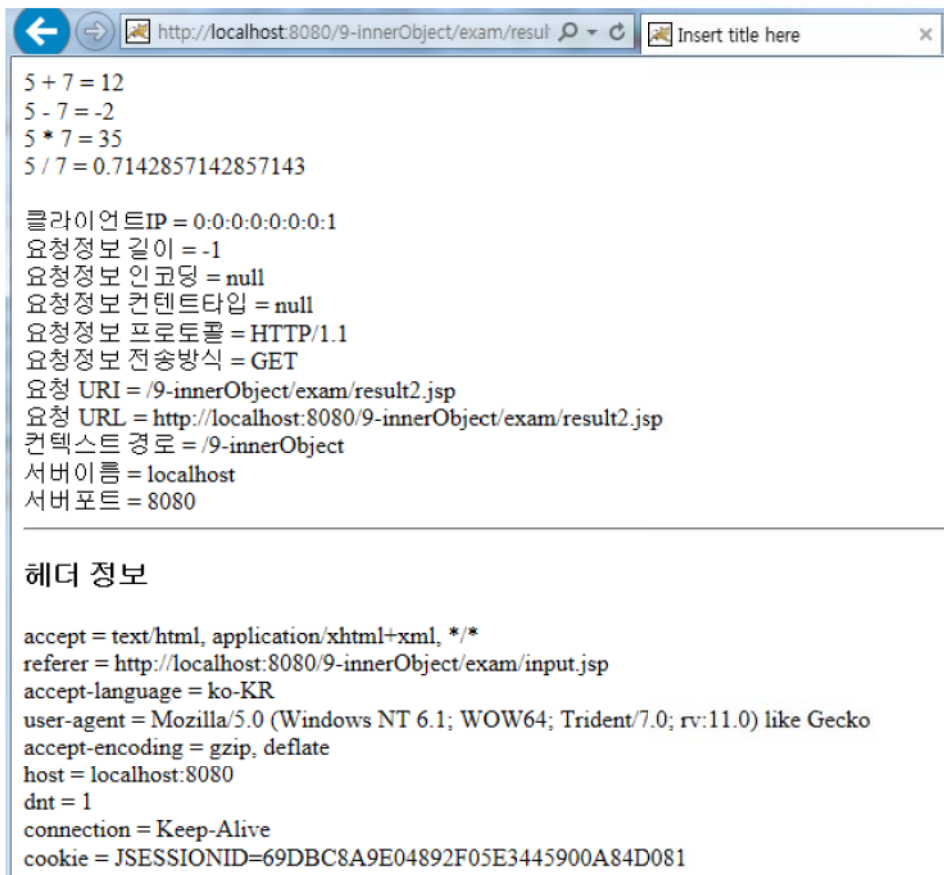
JSP File : WebContent/exam/input.jsp
WebContent/exam/result2.jsp

<실행 결과>

- input.jsp



- result2.jsp



<설명>

URL

http://localhost:8080/9-innerObject/exam/result2.jsp

↓

9-innerObject/exam/result2.jsp => 축약

↓

result2.jsp => 축약

3. response 객체

- 웹 브라우저의 요청에 대한 응답 정보를 저장하고 있는 객체
- 응답 정보와 관련하여 주로 헤더 정보 입력, 리다이렉트 등의 기능을 제공한다.

(1) response 객체의 메소드

① void setHeader(name, value)

- 헤더 정보 값을 수정

② void setContentType(type)

- 웹 브라우저의 요청 결과로 보일 페이지의 contentType을 설정

③ void sendRedirect(url)

- 페이지를 이동

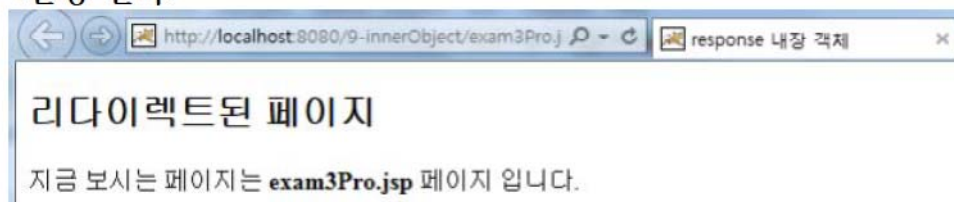
예제3. 서버에서 내부적으로 다른 페이지로 이동하기

Dynamic Web Project : 9-innerObject

JSP File : WebContent/exam3.jsp

WebContent/exam3Pro.jsp

<실행 결과>



4. out 객체

- JSP 페이지의 출력할 내용을 가지고 있는 출력 스트림 객체
- 표현식(<%=문장%>) 과 같다.

(1) out 객체의 메소드

- ① boolean isAutoFlush()
 - 출력 버퍼가 다 찼을 때 처리 여부를 결정
- ② int getBufferSize()
 - 전체 출력 버퍼의 크기를 리턴
- ③ int getRemaining()
 - 현재 남아 있는 출력 버퍼의 크기 리턴
- ④ void clearBuffer()
 - 출력 버퍼에 저장되어 있는 내용을 비움
- ⑤ String println(str)
 - 주어진 내용을 출력. 이때 줄 바꿈은 적용되지 않음
- ⑥ void flush()
 - 출력 버퍼의 내용을 웹 브라우저에 전송하고 비움
- ⑦ void close()
 - 출력 버퍼의 내용을 웹 브라우저에 전송하고 출력 스트림을 닫음

예제4. out 내장 객체를 이용해서 내용 출력하기

Dynamic Web Project : 9-innerObject

JSP File : WebContent/exam4.jsp

<실행 결과>



5. application 객체

- 웹 애플리케이션 Context의 정보를 저장하고 있는 객체
- 서버의 설정 정보, 자원에 대한 정보, 애플리케이션이 실행되는 동안에 발생할 수 있는 이벤트 로그 정보등을 제공
- 웹 애플리케이션 당 1개의 객체가 생성
- 주로 방문자 카운트와 같은 하나의 웹 애플리케이션에서 공유하는 변수에 사용

(1) application 객체 메소드

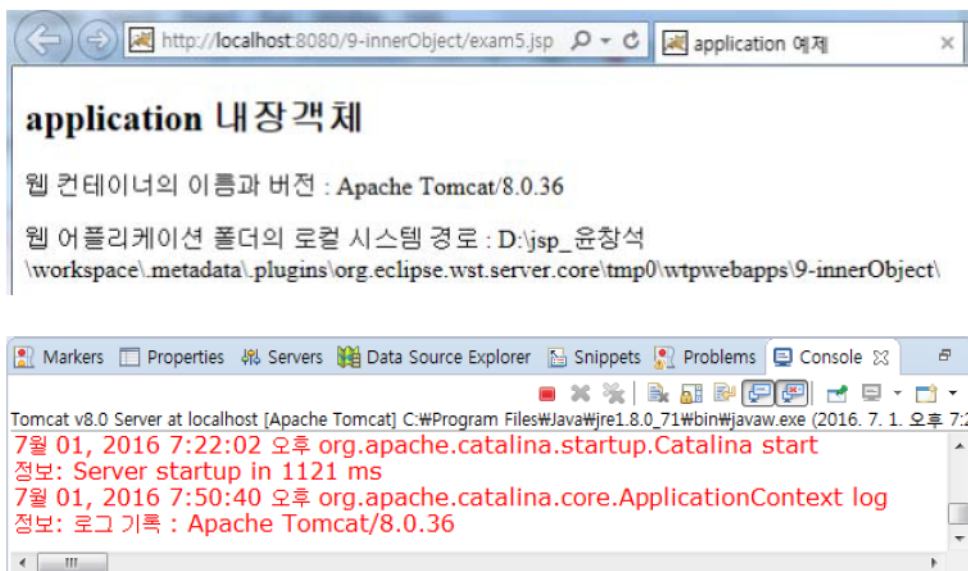
- ① String getServerInfo()
 - 웹 컨테이너의 이름과 버전을 리턴
- ② String getMimeType(fileName)
 - 지정한 파일의 MIME 타입 리턴
- ③ String getRealPath(path)
 - 지정한 경로를 웹 애플리케이션 시스템상의 경로로 변경하여 리턴
- ④ void log(message)
 - 로그 파일에 message를 기록

예제5. application 내장객체 사용하기

Dynamic Web Project : 9-innerObject

JSP File : WebContent/exam5.jsp

<실행 결과>



10장. 회원가입 및 로그인

<작업 순서>

- ① 회원 DB 만들기
- ② 회원가입 폼 페이지 만들기
- ③ 회원가입 데이터 처리 작업
- ④ 회원가입 JDBC 작업
- ⑤ 로그인 폼 페이지 만들기
- ⑥ 회원검색 데이터 처리 작업
- ⑦ 회원 확인 및 로그인 처리
- ⑧ 회원등록시 ID 중복체크하기

1. 회원가입

예제1. 회원 DB 만들기

* Database 계정 jsp에 테이블 만들기

- sqldeveloper 실행하기
- jspexam/m1234 연결하기
- member 테이블 만들기
- 데이터 추가, 검색, 수정, 삭제

① 테이블 작성 및 삭제

- 테이블 생성
- 테이블 구조 확인
- 테이블 삭제

② 데이터 추가

③ 데이터 검색

④ 데이터 수정

⑤ 데이터 삭제

예제2. 회원가입 폼 페이지 만들기

Dynamic Web Project : 10-member

[클라이언트]

Folder : member

JSP File : WebContent/member/writeForm.jsp

Folder : script

Javascript File : WebContent/script/memberScript.js

[서버]

JSP File : WebContent/member/write.jsp

<실행 결과>

A screenshot of a web browser showing a registration form at <http://localhost:8080/10-member/member/writeForm.jsp>. The form fields are: 이름 (이름 입력), 아이디, 비밀번호, 재확인, 성별 (남/여), 이메일 (@naver.com), 핸드폰 (010-), and 주소. Below the form are buttons for '회원가입' and '다시 작성'. A blue alert box titled '웹 페이지 메시지' is displayed on the right, showing a yellow warning icon and the text '이름을 입력하세요' (Please enter a name), with a '확인' (OK) button.

A screenshot of the same registration form, now filled with data. The fields contain: 이름 (홍길동), 아이디 (num1), 비밀번호 (masked with dots), 재확인 (masked with dots), 성별 (남), 이메일 (num1@naver.com), 핸드폰 (010-1234-5678), and 주소 (경기도 수원시). The '회원가입' and '다시 작성' buttons are still visible at the bottom.

A screenshot of a web browser showing the result of the registration at <http://localhost:8080/10-member/member/write.jsp>. The page displays the following information:

이름 : 홍길동
아이디 : num1
비밀번호 : 1234
성별 : 0
이메일 : num1@naver.com
핸드폰 : 010-1234-5678
주소 : 경기도 수원시

* 자바빈 (JavaBean)

(1) 자바빈 규약

- 멤버변수마다 별도의 getter/setter 메소드가 존재해야 한다.
- getter는 파라미터가 존재하지 않아야 한다.
- setter는 반드시 하나 이상의 파라미터가 존재해야 한다.
- 파라미터가 존재하지 않는 기본 생성자를 가진다.
- 멤버변수의 접근 제한자는 private이고, 각 setter/getter는 public으로 정의되어야 하며, 클래스는 public으로 정의한다.

(2) 용도

- 웹페이지를 디자인 부분과 로직 부분을 분리하기 위함이다.
- 여러개 변수를 묶어서 관리하기 위해서이다.

예제3. 회원가입 정보를 데이터베이스에 저장하기

Dynamic Web Project : 10-member

[클라이언트]

Folder : member
JSP File : WebContent/member/writeForm.jsp

Folder : script
Javascript File : WebContent/script/memberScript.js

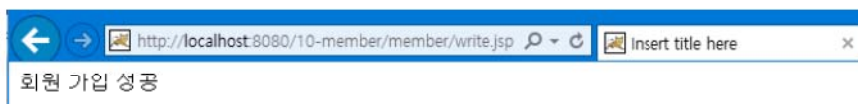
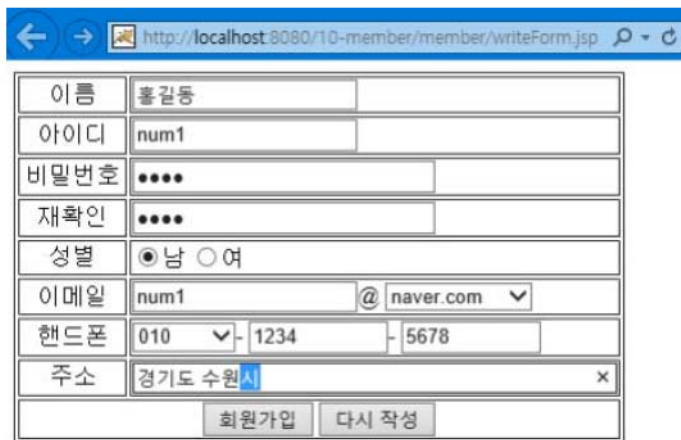
[서버]

JSP File : WebContent/member/write.jsp

Package : member.bean
Class File : src/member.bean/MemberDTO.java
(Data Transfer Object)

Package : member.dao
Class File : src/member.dao/MemberDAO.java
(Data Access Object)

<실행 결과>



워크시트

질의 작성기

-- 데이터 검색

select * from member;

스크립트 출력 x

질의 결과 x

SQL | 인출된 모든 행: 2(0,001초)

| | NAME | ID | PWD | GENDER | EMAIL1 | EMAIL2 | TEL1 | TEL2 | TEL3 | ADDR | LOGTIME |
|---|------|------|------|--------|--------|-----------|------|------|------|---------|----------|
| 1 | 홍길동 | num1 | 1234 | 0 | num1 | naver.com | 010 | 1234 | 5678 | 경기도 수원시 | 16/07/02 |
| 2 | 홍길동 | hong | 1234 | 남 | hong | naver.com | 010 | 1234 | 5678 | 경기도 수원시 | 16/07/02 |

2. 로그인

예제4. 로그인 처리하기

Dynamic Web Project : 10-member

[클라이언트]

Folder : member

JSP File : WebContent/member/loginForm.jsp

Folder : script

Javascript File : WebContent/script/memberScript.js (수정)

[서버]

JSP File : WebContent/member/login.jsp

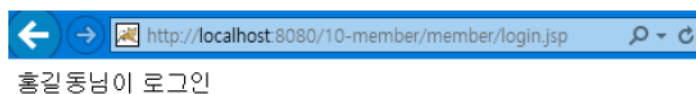
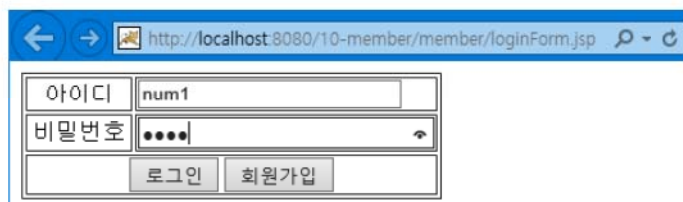
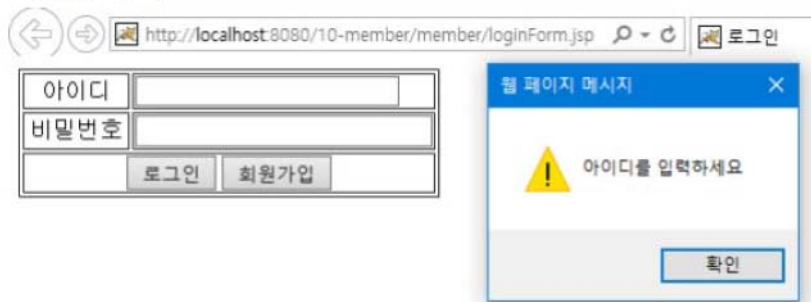
Package : member.bean

Class File : src/member.bean/MemberDTO.java (수정없이 그대로 사용)
(Data Transfer Object)

Package : member.dao

Class File : src/member.dao/MemberDAO.java (수정)
(Data Access Object)

<실행 결과>



3. ID 중복체크

예제5. 회원등록시 ID 중복체크하기

Dynamic Web Project : 10-member

[클라이언트]

Folder : member
JSP File : WebContent/member/writeForm.jsp (수정)
WebContent/member/checkId.jsp

Folder : script
Javascript File : WebContent/script/memberScript.js (수정)

[서버]

Package : member.bean
Class File : src/member.bean/MemberDTO.java (수정없이 그대로 사용)
(Data Transfer Object)

Package : member.dao
Class File : src/member.dao/MemberDAO.java (수정)
(Data Access Object)

<실행 결과>

