

# 3장. 내장 객체 & 폴

*JSP 내장 객체*



# 내장 객체

## 내장 객체(implicit object)

JSP 페이지에서 사용할 수 있도록 JSP 컨테이너에 미리 정의된 객체이다.

- **request** : 웹 브라우저에서 서버의 JSP 페이지로 전달하는 정보를 저장한다.
- **response** : 사용자의 요청을 처리한 결과를 서버에서 웹 브라우저로 전달하는 정보를 저장한다.

### request 객체

메소드	반환유형	설 명
<code>getParameter(String name)</code>	<code>String</code>	요청 파라미터 이름이 name인 값을 전달받음
<code>getParameterValues(String name)</code>	<code>String[ ]</code>	모든 요청 파라미터 이름이 name인 값을 배열 형태로 전달 받음
<code>getParameterNames()</code>	<code>Java.util.Enumeration</code>	모든 요청 파라미터의 이름과 값을 Enumeration 객체 타입으로 전달 받음



# request 객체

## ➤ request 객체를 이용한 이름 등록하기

localhost:8080/jwbook/ch03/request/request.jsp

이름:



localhost:8080/jwbook/ch03/request/process.jsp

이름: 김기용

request/request.jsp

```
<form action="process.jsp" method="post">
  <label for="name">이름 : </label>
  <input type="text" id="name" name="name">
  <input type="submit" value="등록">
</form>
```



# request 객체

request/process.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>
<%
    //폼 페이지에서 전달 받은 한글 인코딩 처리
    request.setCharacterEncoding("utf-8");

    //이름 필드에 입력된 값 가져오기
    String name = request.getParameter("name");

%>
<p>이름: <%= name %></p>
```

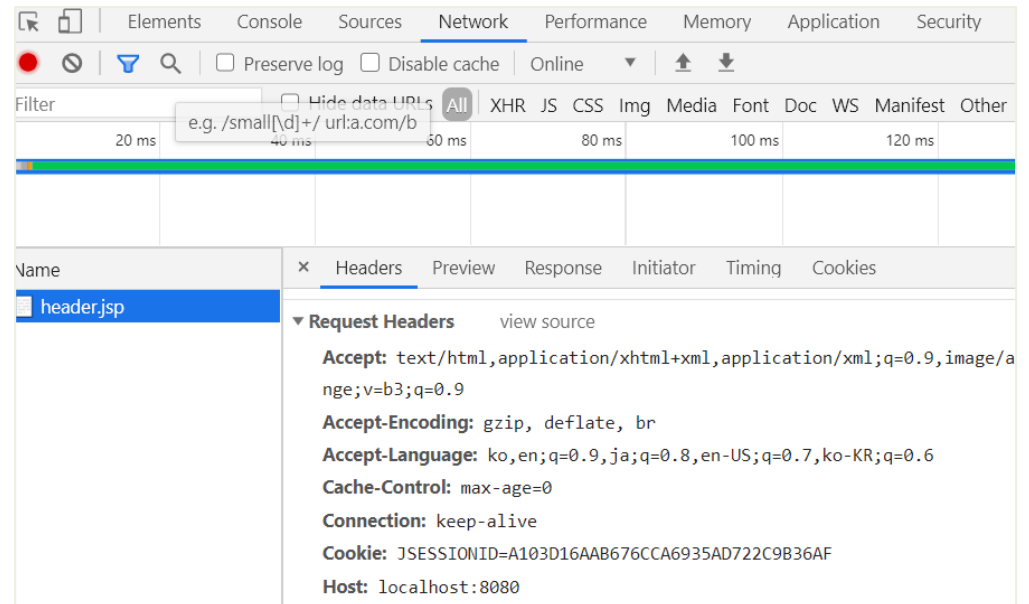
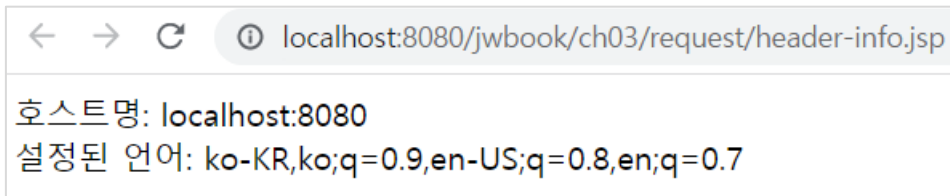


# request 객체

## ➤ HTTP 헤더 정보 값 출력하기

웹 브라우저는 HTTP 헤더에 부가적인 정보를 담아 서버로 전송한다.

request 내장 객체는 헤더 정보나 쿠키 관련 정보를 얻을수 있는 메서드를 제공한다.



# request 객체

## ➤ HTTP 헤더 정보 값 출력하기

request/request2.jsp

```
<%  
    String hostValue = request.getHeader("host");  
    String alValue = request.getHeader("accept-language");  
  
    out.println("호스트명 : " + hostValue + "<br>");  
    out.println("설정된 언어 : " + alValue + "<br>");  
%>
```



# request 객체

## ➤ HTTP 헤더의 모든 정보 값 출력하기

```
localhost:8080/Chapter04/header-info2.jsp

host: localhost:8080
connection: keep-alive
sec-ch-ua: "Chromium";v="88", "Google Chrome";v="88", ";Not
sec-ch-ua-mobile: ?0
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleW
accept: text/html,application/xhtml+xml,application/xml;q=0.9,ir
sec-fetch-site: none
sec-fetch-mode: navigate
sec-fetch-user: ?1
sec-fetch-dest: document
accept-encoding: gzip, deflate, br
accept-language: ko,en;q=0.9,ja;q=0.8,en-US;q=0.7,ko-KR;q=0.6
cookie: JSESSIONID=6E6584C9DA1FC520C077E096DDC053D2
```



# request 객체

## ➤ HTTP 헤더의 모든 정보 값 출력하기

```
<%@page import="java.util.Enumeration"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    //Header 정보 모두 가져오기
    //반복자 객체 얻기
    Enumeration<String> en = request.getHeaderNames();
    while(en.hasMoreElements()){
        String headerName = en.nextElement();    //headerName을 가져와서
        String headerValue = request.getHeader(headerName);    //headerName 값 요청
        out.print(headerName + ":" + headerValue + "<br>");
    }
%>
```





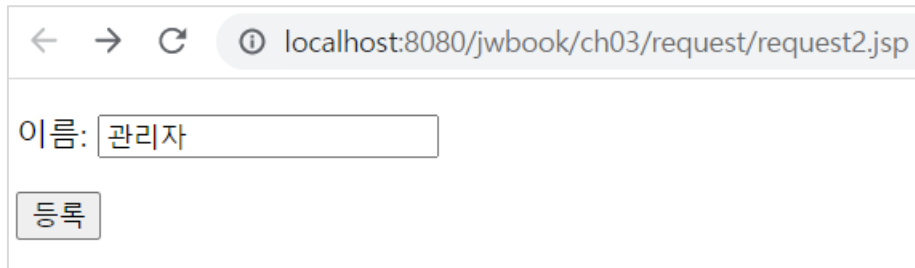
# request 객체

## ➤ 웹 브라우저 및 서버 정보 값 출력하기

request 내장 객체의 웹 브라우저와 서버 관련 메소드를 이용하여 서버 정보 출력

웹 브라우저 정보 – 클라이언트 IP, 요청 정보길이, 전송방식, 요청 URI 등

서버 정보 – 서버 이름과 포트



← → ↻ localhost:8080/jwbook/ch03/request/request2.jsp

이름:

이름 : 관리자  
요청 정보 길이 : 32  
클라이언트 IP : 0:0:0:0:0:0:0:1  
클라이언트 전송방식 : POST  
요청 URI: /Chapter04/process.jsp  
서버 이름 : localhost  
서버 포트 : 8080

# request 객체

## ➤ 웹 브라우저 및 서버 정보 값 출력하기

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

    request.setCharacterEncoding("utf-8");

    String name = request.getParameter("name");
%>
<p>이름: <%= name %><br>
    요청 정보 길이: <%= request.getContentLength() %><br>
    클라이언트 IP: <%= request.getRemoteAddr() %><br>
    클라이언트 전송 방식: <%= request.getMethod() %><br>
    요청 URI: <%= request.getRequestURI() %><br>
    서버 이름: <%= request.getServerName() %><br>
    서버 포트: <%= request.getServerPort() %><br>
</p>
```

request/header-info3.jsp



# response 객체

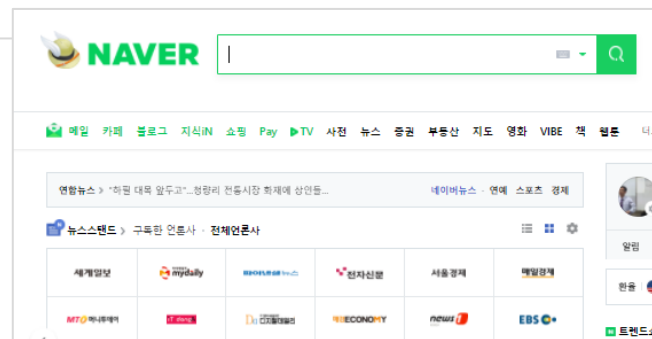
## response 내장 객체

사용자의 요청을 처리한 결과를 서버에서 웹 브라우저로 전달하는 정보를 저장한다  
즉, 서버는 응답 헤더와 요청 처리 결과 데이터를 웹 브라우저로 보낸다.

### 1. 페이지 이동관련 메서드

메소드	반환유형	설 명
sendRedirect(String url)	void	설정한 URL 페이지로 강제 이동합니다.

```
<!DOCTYPE html>
<html>
<head>
    <title>request 객체</title>
</head>
<body>
    <%
        response.sendRedirect("http://www.naver.com");
    %>
</body>
</html>
```



# response 객체

## response 객체로 페이지 이동하기

← → ↻ ⓘ localhost:8080/jwbook/ch03/response/loginForm.jsp

아이디:

패스워드:

← → ↻ ⓘ localhost:8080/jwbook/ch03/response/loginSuccess.jsp

로그인을 성공했습니다.

← → ↻ ⓘ localhost:8080/jwbook/ch03/response/loginFail.jsp

로그인을 실패했습니다.

[로그인 가기](#)



# response 객체

## response 객체로 페이지 이동하기

```
<title>로그인 폼</title>
</head>
<body>
  <form action="LoginProcess.jsp" method="post">
    <p>
      <label for="id">아이디 : </label>
      <input type="text" id="id" name="id">
    </p>
    <p>
      <label for="passwd">비밀번호 : </label>
      <input type="password" id="passwd" name="passwd">
    </p>
    <p><input type="submit" value="전송"></p>
  </form>
</body>
```

loginForm.jsp



# response 객체

## response 객체로 페이지 이동하기

```
<%  
    request.setCharacterEncoding("utf-8");  
  
    String id = request.getParameter("id");  
    String pwd = request.getParameter("passwd");  
  
    if(id.equals("admin") && pwd.equals("1234")){  
        //out.println("로그인을 성공했습니다.");  
        response.sendRedirect("loginSuccess.jsp");  
    }  
    else{  
        //out.println("로그인을 실패했습니다.");  
        response.sendRedirect("loginFail.jsp");  
    }  
%>
```

response/loginProcess.jsp

```
<body>  
    <p>로그인을 성공했습니다.</p>  
</body>
```

response/loginSuccess.jsp

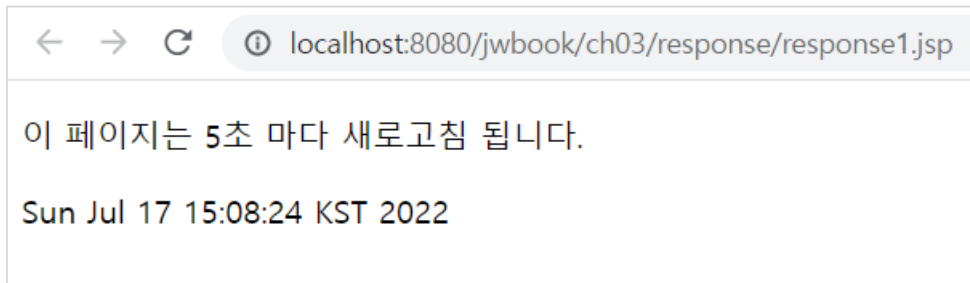
```
<body>  
    <p>로그인을 실패했습니다.</p>  
    <a href="loginForm.jsp">로그인 가기</a>  
</body>
```

response/loginFail.jsp



# response 객체

## response 객체로 5초마다 JSP 페이지 갱신하기

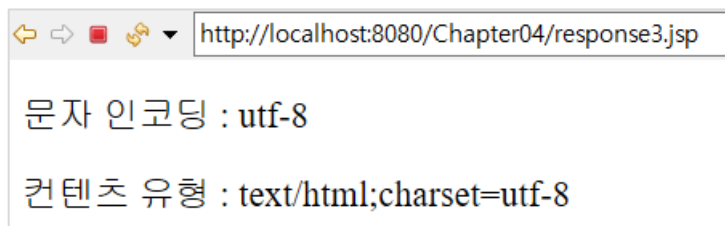


```
<body>
  <p>이 페이지는 5초 마다 새로고침 됩니다.</p>
  <%
    response.setHeader("Refresh", 5);
  %>
  <p><%= new Date() %>
</body>
```



# 내장 객체

## response 객체로 응답 콘텐츠 설정하기



```
<%  
    response.setCharacterEncoding("uft-8");  
    response.setContentType("text/html; charset=utf8");  
%>  
<p> 문자 인코딩 : <%=response.getCharacterEncoding()%>  
<p> 컨텐츠 유형 : <%=response.getContentType() %>
```



# 내장 객체

## 계산기 프로그램 구현

localhost:8080/jwbook/calculator/calculator.jsp

계산기

10

+ ▼

20

계산

다시입력

localhost:8080/jwbook/calculator/result.jsp

계산기

계산 결과 : 30

# JSP 빈즈 프로그래밍

## 계산기 프로그램 - 자바 빈즈로 구현

### 프로그램 소스 목록

파일 이름	역 할
calculator.jsp	계산기 메인 화면
calc.css	전체 레이아웃 등의 스타일을 구현
result.jsp	calculator.jsp에 입력된 내용을 Calculator 빈을 이용해 계산 기능 수행 (import해서 객체 생성)
result2.jsp	calculator.jsp에 입력된 내용을 Calculator 빈을 이용해 계산 기능 수행 (useBean 태그로 구현)
Calculator.java	계산기 자료형을 정의한 빈즈 클래스



# JSP 빈즈 프로그래밍

## 계산기 프로그램 - 자바 빈즈로 구현

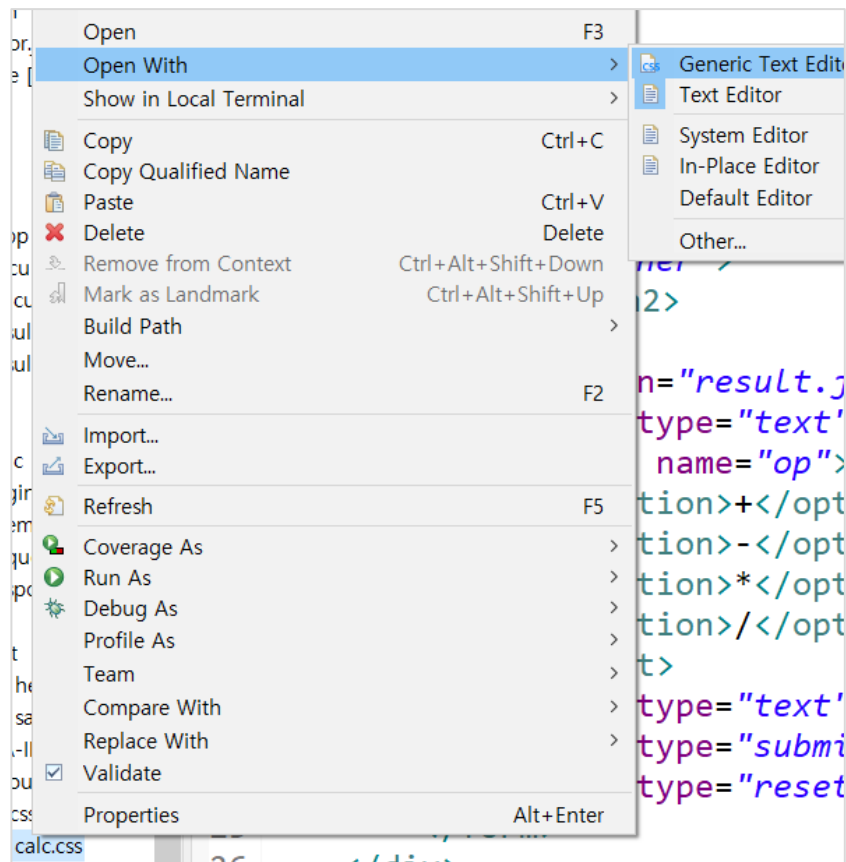
```
<title>계산기 프로그램</title>
<link rel="stylesheet" href="../../resources/css/calc.css">
</head>
<body>
    <div id="container">
        <h2>계산기</h2>
        <hr>
        <form action="result.jsp" method="post" name="form1">
            <input type="text" name="num1">
            <select name="op">
                <option>+</option>
                <option>-</option>
                <option>*</option>
                <option>/</option>
            </select>
            <input type="text" name="num2">
            <input type="submit" value="계산">
            <input type="reset" value="다시입력">
        </form>
    </div>
</body>
```

claculator.jsp



# JSP 빈즈 프로그래밍

## 계산기 프로그램 - 자바 빈즈로 구현



calc.css

```
@charset "UTF-8";

#container{
    width: 800px;
    margin: 0 auto;
    text-align: center;
}

p{font-size: 1.2em;}
```

# JSP 빈즈 프로그래밍

## 계산기 프로그램 - 자바 빈즈로 구현

```
package com.bean;

import java.io.Serializable;

public class Calculator implements Serializable {

    private static final long serialVersionUID = 100000L;

    private int num1 = 0;
    private int num2 = 0;
    private String op = "";
    private int result = 0;

    public int calculate() {
        if(op.equals("+")) {
            result = num1 + num2;
        } else if(op.equals("-")) {
            result = num1 - num2;
        } else if(op.equals("*")) {
            result = num1 * num2;
        } else if(op.equals("/")) {
            result = num1 / num2;
        }
        return result;
    }
}
```

Claculator.java

```
public int getNum1() {
    return num1;
}

public void setNum1(int num1) {
    this.num1 = num1;
}

public int getNum2() {
    return num2;
}

public void setNum2(int num2) {
    this.num2 = num2;
}

public String getOp() {
    return op;
}

public void setOp(String op) {
    this.op = op;
}

public int getResult() {
    return result;
}
```



# JSP 빈즈 프로그래밍

Calculator클래스 import해서 구현

result.jsp

```
<div id="container">
  <h2>계산기</h2>
  <hr>
  <%
    int num1 = Integer.parseInt(request.getParameter("num1"));
    int num2 = Integer.parseInt(request.getParameter("num2"));
    String op = request.getParameter("op");

    Calculator calc = new Calculator();
    calc.setNum1(num1);
    calc.setNum2(num2);
    calc.setOp(op);

    calc.calculate();
  %>
  <p>계산 결과 : <%=calc.getResult() %>
  <hr>
</div>
```



# JSP 빈즈 프로그래밍

useBean 태그로 구현하기

result2.jsp

```
<body>
<jsp:useBean id="calc" class="com.bean.Calculator" />
<jsp:setProperty property="num1" name="calc"/>
<jsp:setProperty property="num2" name="calc"/>
<jsp:setProperty property="op" name="calc"/>
  <div id="container">
    <h2>계산기</h2>
    <hr>
    <%
      calc.calculate();
    %>

    <p>계산 결과 : <%=calc.getResult() %>
    <hr>
  </div>
</body>
```



# Get과 Post 방식

GET 요청과 쿼리 스트링(입력)

`http://localhost/sayHello.jsp`

`http://localhost/sayHello.jsp?cnt=3`

GET

안녕하세요  
안녕하세요  
안녕하세요

localhost:8181/Chapter09/sayHello.jsp

안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP  
안녕 JSP

```
<%  
    for(int i=0; i<10; i++){  
        out.println("안녕 JSP<br>");  
    }  
%>
```





# Get과 Post 방식

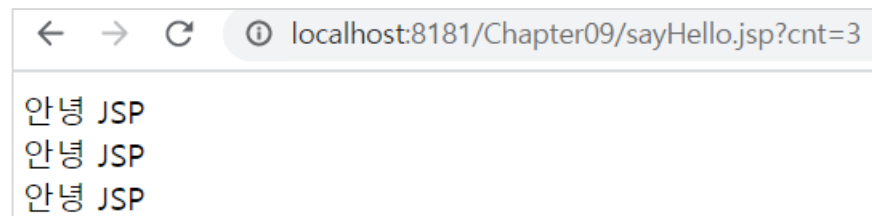
## GET 요청과 쿼리 스트링(입력)

http://localhost/sayHello.jsp?

GET

쿼리스트링이 null인 경우

```
<%  
    int cnt = Integer.parseInt(request.getParameter("cnt"));  
  
    for(int i=0; i<cnt; i++){  
        out.println("안녕 JSP<br>");  
    }  
%>
```



# Get과 Post 방식

## GET 요청과 쿼리 스트링(입력) 오류 해결

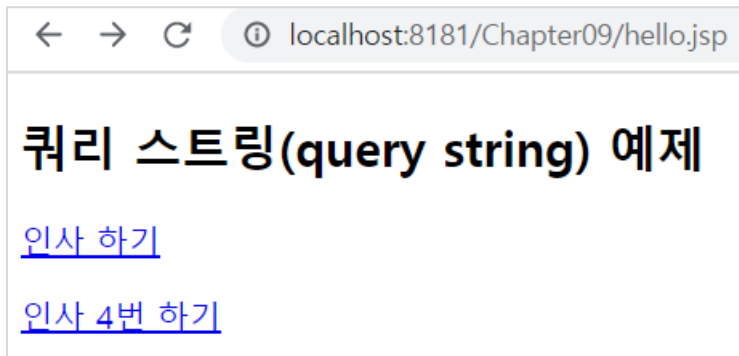
get/sayHello.jsp

```
<%  
    int cnt = 10;  
    if(request.getParameter("cnt") != null){  
        cnt = Integer.parseInt(request.getParameter("cnt"));  
    }  
  
    for(int i=0; i<cnt; i++){  
        out.println("안녕 JSP<br>");  
    }  
%>
```



# Get과 Post 방식

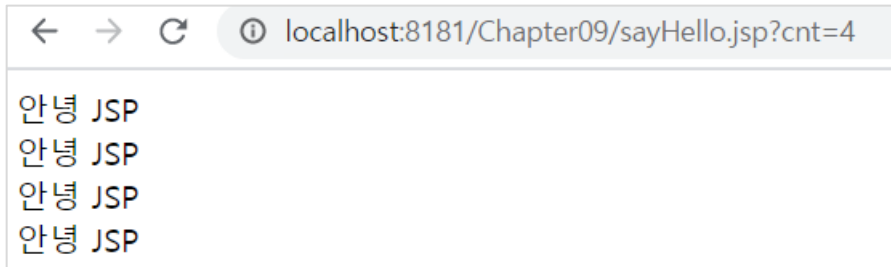
## JSP에서 쿼리 스트링 링크하기



get/hello.jsp

```
<h2>쿼리 스트링(query string) 예제</h2>
<p><a href="sayHello.jsp">인사 하기</a></p>

<p><a href="sayHello.jsp?cnt=4">인사 4번 하기</a></p>
```



# 폼 데이터 처리

## 폼 데이터 처리

joinForm.jsp

```
<h3>회원 가입</h3>
<form action="joinProcess.jsp" method="post">
  <p>
    <label for="id">아이디 : </label>
    <input type="text" id="id" name="id">
  </p>
  <p>
    <label for="passwd">비밀번호 : </label>
    <input type="password" id="passwd" name="passwd">
  </p>
  <p>
    <label for="name">이름 : </label>
    <input type="text" id="name" name="name">
  </p>
  <p>
    <label for="phone">연락처 : </label>
    <input type="text" id="phone1" name="phone1" maxlength=4 size=4>
    - <input type="text" id="phone2" name="phone2" maxlength=4 size=4>
    - <input type="text" id="phone3" name="phone3" maxlength=4 size=4>
  </p>
</form>
```



# 폼 데이터 처리

## 폼 데이터 처리

joinForm.jsp

```
<p>
  <label for="gender">성 별 : </label>
  <input type="radio" id="gender" name="gender" value="남성" checked>남성
  <input type="radio" id="gender" name="gender" value="여성">여성
</p>
<p>
  <label for="hobby">취미 : </label>
  <input type="checkbox" id="hobby1" name="hobby1" value="독서" checked>독서
  <input type="checkbox" id="hobby2" name="hobby2" value="운동">운동
  <input type="checkbox" id="hobby3" name="hobby3" value="영화">영화
</p>
<p>
  <input type="submit" value="가입하기">
  <input type="reset" value="다시쓰기">
</p>
<p><textarea name="comment" cols="30" rows="3"></textarea></p>
</form>
```



# 폼 데이터 처리

## 폼 데이터 처리

### 회원 가입

아이디:

비밀번호:

이름:

연락처:  -  -

성 별: ☐ 남성 ☒ 여성

취 미: ☐ 독서 ☒ 운동 ☒ 영화

안녕하세요~  
국가대표 양궁 선수 안산입니다.

가입하기

다시쓰기

아이디: san1004

비밀번호: 2022

이름: 안산

연락처: 010 - 1234 - 5678

성 별: 여성

취 미: null 운동 영화

가입인사: 안녕하세요~ 국가대표 양궁 선수 안산입니다.

# 폼 데이터 처리

## 폼 데이터 처리

JoinProcess.jsp

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

    String id = request.getParameter("id");
    String passwd = request.getParameter("passwd");
    String name = request.getParameter("name");
    String phone1 = request.getParameter("phone1");
    String phone2 = request.getParameter("phone2");
    String phone3 = request.getParameter("phone3");
    String gender = request.getParameter("gender");
    String hobby1 = request.getParameter("hobby1");
    String hobby2 = request.getParameter("hobby2");
    String hobby3 = request.getParameter("hobby3");
    String comment = request.getParameter("comment");
%>
<p>아이디 : <%=id %>
<p>비밀번호 : <%=passwd %>
<p>이름 : <%=name %>
<p>연락처 : <%=phone1 %> - <%=phone2 %> - <%=phone3 %>
<p>성별 : <%=gender %>
<p>취미 : <%=hobby1 %> <%=hobby2 %> <%=hobby3 %>
<p>가입인사 : <%=comment %>
```



# 폼 데이터 처리

## Hobby 항목 – 배열 : getParameterValues() 메서드

```
<p>
  <label for="gender">성 별 : </label>
  <input type="radio" id="gender" name="gender" value="남성" checked>남성
  <input type="radio" id="gender" name="gender" value="여성">여성
</p>
<p>
  <label for="hobby">취미 : </label>
  <input type="checkbox" id="hobby1" name="hobby" value="독서" checked>독서
  <input type="checkbox" id="hobby2" name="hobby" value="운동">운동
  <input type="checkbox" id="hobby3" name="hobby" value="영화">영화
</p>
<p>
  <input type="submit" value="가입하기">
  <input type="reset" value="다시쓰기">
</p>
<p><textarea name="comment" cols="30" rows="3"></textarea></p>
</form>
```



# 폼 데이터 처리

## 폼 데이터 처리

아이디: san1004

비밀번호: 12345

이름: 안산

연락처: 010 - 1234 - 5678

성별: 여성

취미: 운동 영화

가입인사: 안녕하세요~ 국가대표 양궁

```
<%
```

```
//한글 인코딩 처리
```

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

```
//폼의 입력값을 전달 받음
```

```
String id = request.getParameter("userid");
```

```
String pw = request.getParameter("passwd");
```

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

```
String phone1 = request.getParameter("phone1");
```

```
String phone2 = request.getParameter("phone2");
```

```
String phone3 = request.getParameter("phone3");
```

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

```
String[] hobby = request.getParameterValues("hobby");
```

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

```
%>
```

```
<p>아이디 : <%=id %></p>
```

```
<p>패스워드 : <%=pw %></p>
```

```
<p>이름 : <%=name %></p>
```

```
<p>연락처 : <%=phone1 %> - <%=phone2 %> - <%=phone3 %> </p>
```

```
<p>성별 : <%=gender %></p>
```

```
<p>취미 : <% for(int i=0; i<hobby.length; i++){  
    out.println(hobby[i]);  
}
```

```
%> </p>
```

```
<p>가입인사 : <%=comment %></p>
```



# 폼 데이터 처리

## 다중 선택 처리 실습 문제

폼 자료 처리 x +

← → ↻ ⓘ localhost:8181/Chapter05/exercise/form03.jsp

오렌지 ☒ 사과 ☐ 바나나 ☒ 전송

```
<form action="form03_process.jsp" method="post">
  <p>
    오렌지<input type="checkbox" name="fruit" value="Orange" checked>
    사과<input type="checkbox" name="fruit" value="Apple" >
    바나나<input type="checkbox" name="fruit" value="Banana" >
    <input type="submit" value="전송">
  </p>
</form>
```

# 폼 데이터 처리

## 다중 선택 처리 실습 문제

← → ↻ ⓘ localhost:8181/Chapter05/exercise/form03\_process.jsp

선택한 과일

Orange Banana

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

    String[] fruit = request.getParameterValues("fruit");
%>
<p>선택한 과일
<p> <%
    for(int i=0; i<fruit.length; i++){
        out.println(fruit[i]);
    }
%>
</p>
```

