



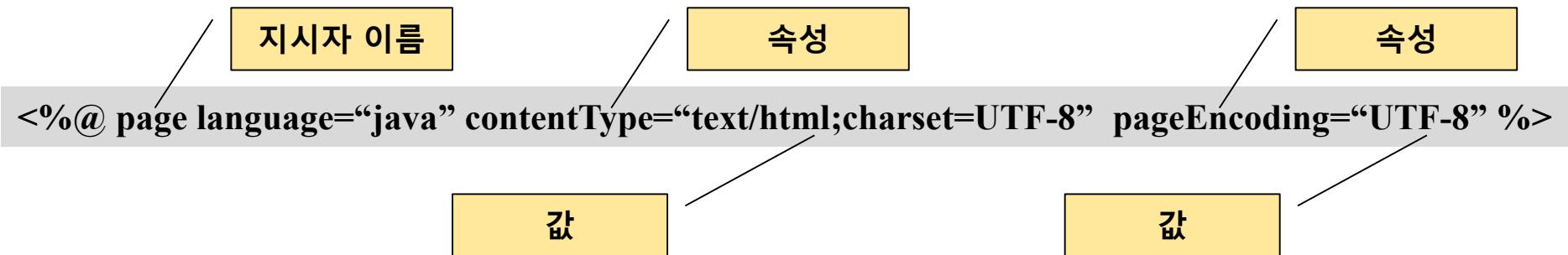
03. JSP 구성 요소

JSP 구성 요소

- 디렉티브(Directive) – 지시자
- 스크립트
 - 스크립틀릿(scriptlet)
 - 표현식(expression)
 - 선언부(declaration)
- 기본 객체

Directive(디렉티브, 지시자)

- JSP 페이지에 대한 정보를 설정할 때 사용
 - <%@ 지시자이름 속성="값" 속성="값" ... %>의 형태로 작성



- 제공되는 지시자
 - page** : JSP 페이지에 대한 정보를 지정하며 문서의 타입, 출력 버퍼의 크기, 에러 페이지 등의 정보를 입력
 - include** : 특정 영역에 다른 문서를 포함
 - taglib** : 태그 라이브러리를 지정

page 지시자

- **역할**

- JSP 페이지가 어떤 문서를 생성하는지
- 어떤 자바 클래스를 사용하는지
- 세션에 참여하는지
- 출력 버퍼의 존재 여부 등을 알 수 있다.

page 지시자 속성

속성	설명	기본 값
language	현재 JSP 페이지가 사용할 프로그래밍 언어를 설정합니다.	java
contentType	현재 JSP 페이지가 생성할 문서의 콘텐츠 유형을 설정합니다.	text/html
pageEncoding	현재 JSP 페이지의 문자 인코딩을 설정합니다.	ISO-8859-1
import	현재 JSP 페이지가 사용할 자바 클래스를 설정합니다.	
session	현재 JSP 페이지의 세션 사용 여부를 설정합니다.	true
buffer	현재 JSP 페이지의 출력 버퍼 크기를 설정합니다.	8KB

page 지시자 속성

속성	설명	기본 값
autoFlush	출력 버퍼의 동작 제어를 설정합니다.	true
isThreadSafe	현재 JSP 페이지의 멀티스레드 허용 여부를 설정합니다.	true
info	현재 JSP 페이지에 대한 설명을 설정합니다.	
errorPage	현재 JSP 페이지에 오류가 발생했을 때 보여줄 오류 페이지를 설정합니다.	
isErrorPage	현재 JSP 페이지가 오류 페이지인지 여부를 설정합니다.	false
isELIgnored	현재 JSP 페이지의 표현 언어(EL) 지원 여부를 설정합니다.	false
isScriptingEnabled	현재 JSP 페이지의 스크립트 태그 사용 여부를 설정합니다.	

page 지시자 속성 실습

- **import** 속성 – 스크립트 요소 안에서 사용할 자바 클래스와 인터페이스를 임포트 하기 위해 사용하는 속성

<dateTime.jsp>

```
2 <%@page import="java.text.SimpleDateFormat"%>
3 <%@page import="java.util.Date"%>
4 <%@ page language="java" contentType="text/html; charset=UTF-8"
5     pageEncoding="UTF-8"%>
```

```
<body>
<%
    Date date = new Date();
    SimpleDateFormat sdfDate = new SimpleDateFormat("yyyy년 MM월 dd일입니다.");
    SimpleDateFormat sdfTime = new SimpleDateFormat("hh시 mm분 ss초");
    String dateStr = sdfDate.format(date);
    String timeStr = sdfTime.format(date);
%>
<h1>오늘은 <%= dateStr %></h1>
<h2>현재 시각 : <%=timeStr %></h2>

</body>
```

include 지시자

- **특징**

- JSP 페이지의 특정 영역에 다른 문서를 포함시킨다.
- 불러올 대상은 **file 속성을 이용**해서 지정한다.
- 지정된 속성 값은 지시자가 속하는 JSP 페이지를 기준으로 한 상대적인 URL로 해석된다.

현재 디렉터리에 있는 `dateTime.jsp` 를 include

```
<%@ include file="dateTime.jsp" %>
```

```
<%@ include file="sub1(dateTime.jsp" %>
```

하위 `sub1` 디렉터리에 있는 `dateTime.jsp` 를 include

include 지시자 실습

- include 지시자의 사용 예

< include.jsp >

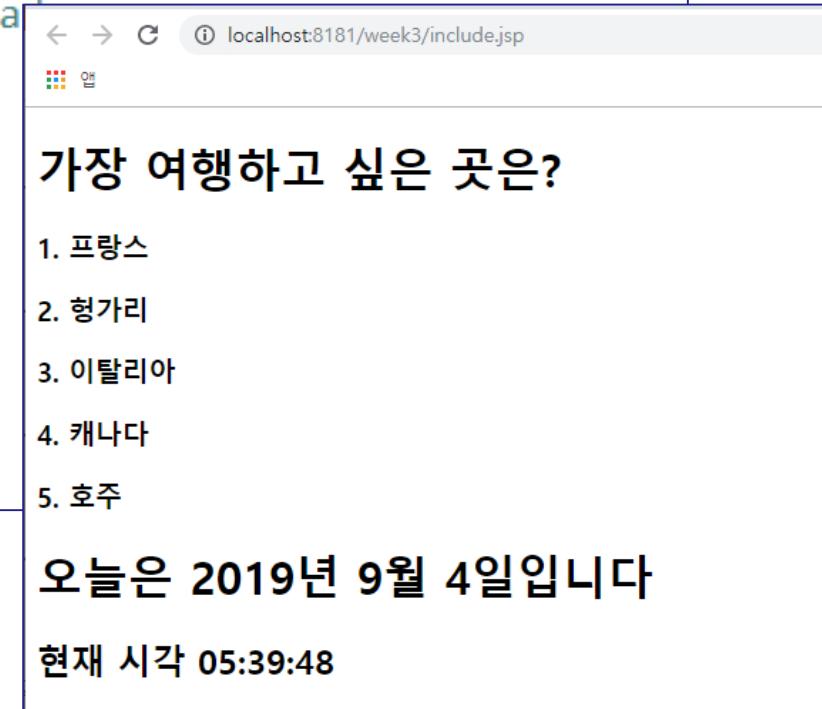
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3<!DOCTYPE html><html><head><meta charset="UTF-8">
4 <title>Insert title here</title></head>
5<body>
6 <h1>가장 여행하고 싶은 곳은?</h1>
7 <h3>1. 프랑스</h3>
8 <h3>2. 헝가리</h3>
9 <h3>3. 이탈리아</h3>
10 <h3>4. 캐나다</h3>
11 <h3>5. 호주</h3>
12 <%@ include file="dateTime.jsp" %>
13 </body></html>
```

include 지시자 실습

- include 지시자의 사용 예

< include.jsp >

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3<!DOCTYPE html><html><head><meta charset="UTF-8">
4 <title>Insert title here</title></head>
5<body>
6 <h1>가장 여행하고 싶은 곳은?</h1>
7 <h3>1. 프랑스</h3>
8 <h3>2. 헝가리</h3>
9 <h3>3. 이탈리아</h3>
10 <h3>4. 캐나다</h3>
11 <h3>5. 호주</h3>
12 <%@ include file="dateTime.jsp" %>
13 </body></html>
```



include 지시자 실습

- include 지시자의 사용 예

< include.jsp 페이지 소스 보기 결과 >

The screenshot shows the browser's developer tools with the 'View Source' tab selected. The URL in the address bar is 'view-source:localhost:8181/week3/include.jsp'. The content area displays the rendered HTML code, which includes the main page structure and the included 'dateTime.jsp' file.

```
1 <%@ page language="java" %>
2     pageEncoding="UTF-8"%>
3<!DOCTYPE html><html><head>
4 <title>Insert title here</title>
5<body>
6 <h1>가장 여행하고 싶은 곳은?</h1>
7 <h3>1. 프랑스</h3>
8 <h3>2. 헝가리</h3>
9 <h3>3. 이탈리아</h3>
10 <h3>4. 캐나다</h3>
11 <h3>5. 호주</h3>
12 <%@ include file="dateTime.jsp" %>
13 </body></html>

1 <!DOCTYPE html><html><head><meta charset="UTF-8">
2 <title>Insert title here</title></head>
3 <body>
4 <h1>오늘은 2019년
5         9월
6         4일입니다</h1>
7 <h2>현재 시각 05:39:48</h2>
8 </body></html>
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
```

JSP 구성 요소

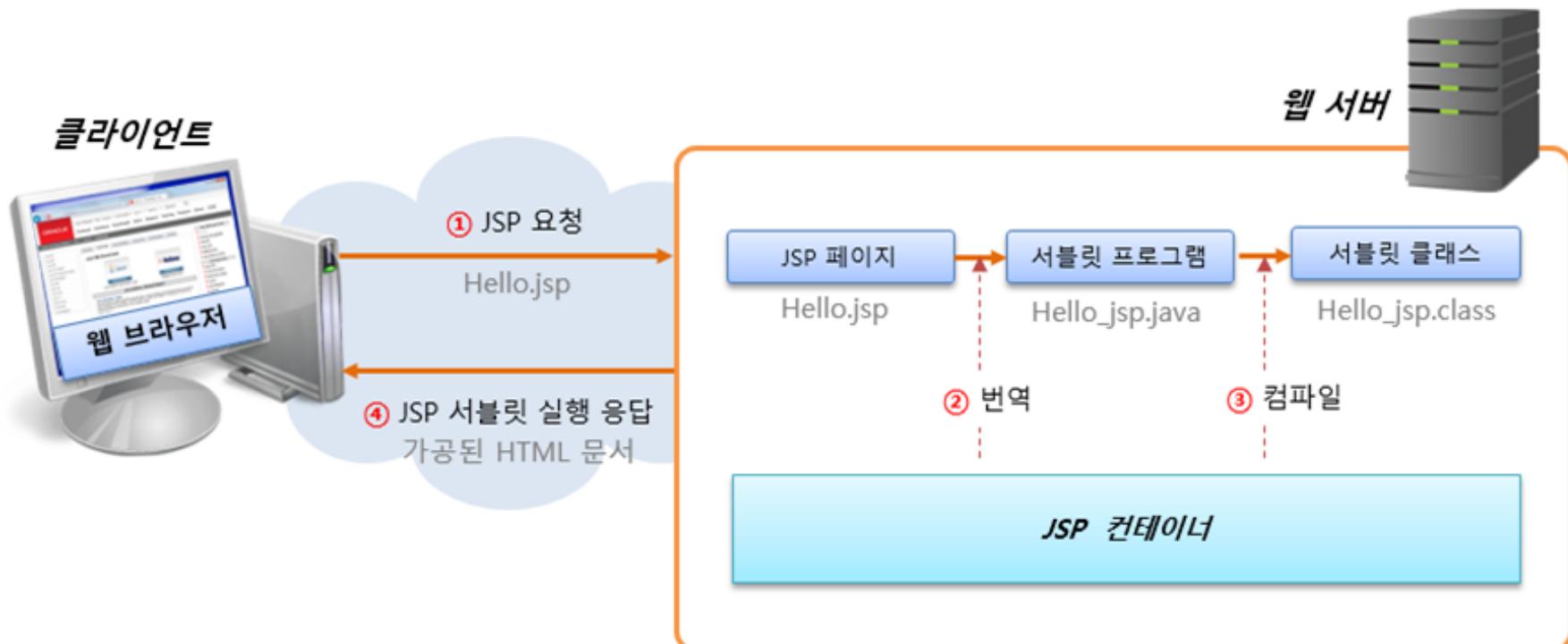
- 디렉티브(Directive) – 지시자
- **스크립트**
 - **스크립틀릿(scriptlet)**
 - **표현식(expression)**
 - **선언부(declaration)**

스크립트 구성 요소

- **스크립틀릿(scriptlet)**
 - <% ... %> 로 기술
 - 프로그래밍 코드 기술에 사용
- **표현식(expression)**
 - <%= ... %> 로 기술
 - 화면에 출력할 내용 기술에 사용
- **선언문(declaration)**
 - <%! ... %> 로 기술
 - 전역변수 선언 및 메소드 선언에 사용

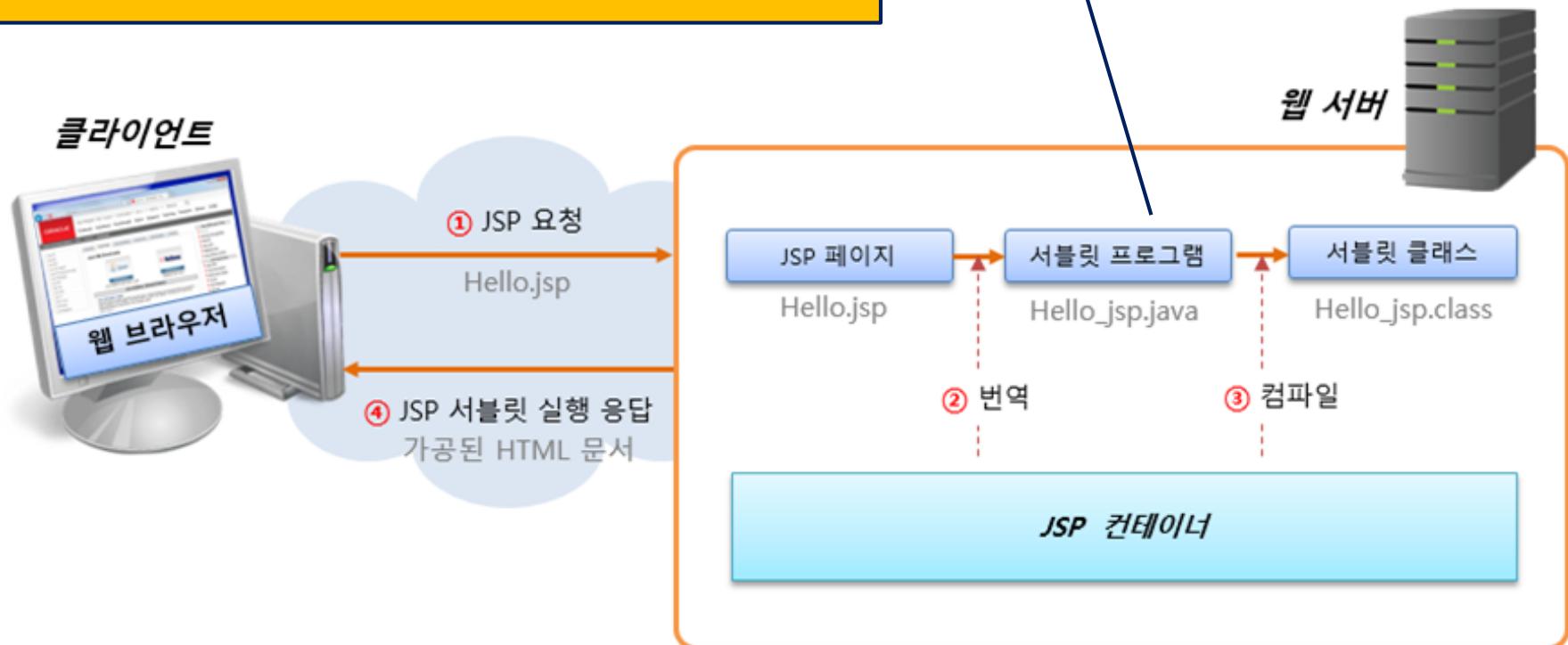
스크립트 구성 요소

- 스크립틀릿(scriptlet)
- 표현식(expression)
- 선언문(declaration)



스크립트 구성 요소

생성된 자바 소스 코드 & 서블릿 클래스 위치 :
workspace 위치 \\ .metadata \\ .plugins \\ org.eclipse.wst.server.core \\ tmp0 \\ work \\ Catalina \\ localhost \\ week3 \\ work \\ apache \\ jsp



JSP 페이지 주석

- 주석을 기술하는 방법
 - JSP 페이지에 주석을 다는 방법은 다양하다

- 페이지 내 **HTML 코드** 부분 주석

```
<!-- HTML의 주석 -->
```

- JSP 페이지 내 **스크립틀릿** 부분 주석

```
/* JAVA의 주석 */
```

```
// JAVA의 주석
```

- **JSP 고유의 주석**

```
<%-- JSP의 주석 --%>
```

JSP 페이지 주석 실습

<comment.jsp>

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html><html><head><meta charset="UTF-8">
4 <title>Insert title here</title></head>
5 <body>
6     <!-- 이 페이지는 JSP에 의해 생성된 HTML 문서입니다 -->
7
8 <%
9     int result = 1;          //곱셈 결과를 저장하는 변수
10
11    /* 1부터 10까지 곱하는 반복문 */
12    for (int cnt=1; cnt<=10; cnt++){
13        result *= cnt;
14    }
15 <%
16 <br>
17    1부터 10까지 곱한 값은 ? <%-- <%= result %> --%>
18 </body></html>
```

JSP 페이지 주석

- JSP 페이지 실행 후 소스보기
 - HTML 주석만 보여준다
- 주석의 활용
 - 주석의 용도에 따라 외부 노출이 가능한 것은 => HTML 주석 사용
 - 외부에 노출되면 안되는 주석의 경우 => JSP 주석 사용
- 주석 단축키
 - **CTRL+SHIFT+/-**

JSP 구성 요소

- 디렉티브(Directive) – 지시자
- 스크립트
 - 스크립틀릿(scriptlet)
 - 표현식(expression)
 - 선언부(declaration)
- 기본 객체



JSP 기본 객체

JSP가 제공하는 기본 객체

- JSP는 웹 어플리케이션 프로그래밍을 하는데 필요한 기능을 제공
 - 기본 객체(implicit object)

기본 객체	실제 타입	설 명
request	javax.servlet.http.HttpServletRequest	클라이언트의 요청 정보 를 저장한다
response	javax.servlet.http.HttpServletResponse	응답 정보 를 저장한다
pageContext	javax.servlet.jsp.PageContext	JSP 페이지에 대한 정보 를 저장한다
session	javax.servlet.http.HttpSession	HTTP 세션 정보 를 저장한다
application	javax.servlet.ServletContext	웹 어플리케이션에 대한 정보 를 저장한다
out	javax.servlet.jsp.JspWriter	JSP 페이지가 생성하는 결과를 출력할 때 사용하는 출력 스트림 이다
config	javax.servlet.ServletConfig	JSP 페이지에 대한 설정 정보 를 저장한다
page	java.lang.Object	JSP 페이지를 구현한 자바 클래스 인스턴스이다
exception	java.lang.Throwable	익셉션 객체이다. 에러 페이지에서만 사용한다

request 기본 객체

- 속성
 - JSP 페이지에서 가장 많이 사용되는 기본 객체
 - 웹 브라우저의 요청 정보를 제공
- 기능
 - 클라이언트(웹 브라우저)와 관련된 정보 읽기
 - 서버와 관련된 정보 읽기
 - 클라이언트가 전송한 요청 파라미터 읽기
 - 클라이언트가 전송한 쿠키 읽기
 - 속성 처리

request 기본 객체

- 클라이언트 및 서버 정보 관련 메서드

메서드	리턴 타입	설명
getRemoteAddr()	String	웹 서버에 연결한 클라이언트의 IP 주소 를 구한다
getContentLength()	long	클라이언트가 전송한 요청 정보의 길이 를 구한다
getCharacterEncoding()	String	클라이언트가 요청 정보를 전송할 때 사용한 캐릭터의 인코딩 을 구한다
getContentType()	String	클라이언트가 요청 정보를 전송할 때 사용한 컨텐츠의 타입 을 구한다
getProtocol()	String	클라이언트가 요청한 프로토콜 을 구한다
getMethod()	String	웹 브라우저가 정보를 전송할 때 사용한 요청방식 을 구한다
getRequestURL()	String	웹 브라우저가 요청한 URL에서 경로 를 구한다
getContextPath()	String	JSP 페이지가 속한 웹 어플리케이션의 컨텍스트 경로 를 구한다
getServerName()	String	연결할 때 사용한 서버 이름 을 구한다
getServerPort()	int	서버가 실행 중인 포트 번호 를 구한다

request 기본 객체 정보 실습

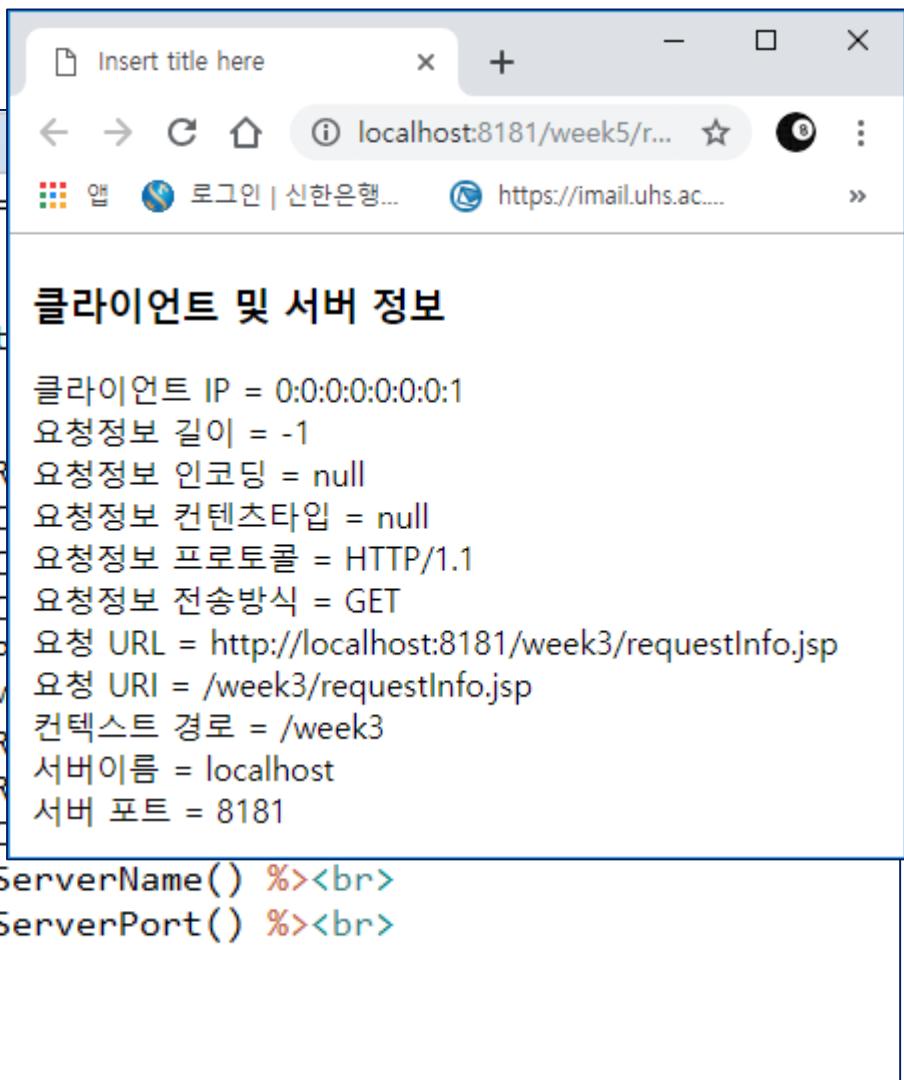
- requestInfo.jsp

```
requestInfo.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3<!DOCTYPE html><html><head>
4 <meta charset="UTF-8"><title>Insert title here</title></head>
5<body>
6     <h3>클라이언트 및 서버 정보</h3>
7     클라이언트 IP          = <%=request.getRemoteAddr() %><br>
8     요청정보 길이          = <%=request.getContentLength() %><br>
9     요청정보 인코딩        = <%=request.getCharacterEncoding() %><br>
10    요청정보 컨텐츠타입      = <%=request.getContentType() %><br>
11    요청정보 프로토콜      = <%=request.getProtocol() %><br>
12    요청정보 전송방식      = <%=request.getMethod() %><br>
13    요청 URL              = <%=request.getRequestURL() %><br>
14    요청 URI               = <%=request.getRequestURI() %><br>
15    컨텍스트 경로          = <%=request.getContextPath() %><br>
16    서버이름              = <%=request.getServerName() %><br>
17    서버 포트              = <%=request.getServerPort() %><br>
18 </body>
19 </html>
20
```

request 기본 객체 정보 실습

- requestInfo.jsp

```
requestInfo.jsp
1 <%@ page language="java" contentType=
2   pageEncoding="UTF-8"%>
3<!DOCTYPE html><html><head>
4 <meta charset="UTF-8"><title>Insert t
5<body>
6   <h3>클라이언트 및 서버 정보</h3>
7   클라이언트 IP      = <%=request.getR
8   요청정보 길이       = <%=request.getO
9   요청정보 인코딩     = <%=request.getO
10  요청정보 컨텐츠타입 = <%=request.getO
11  요청정보 프로토콜   = <%=request.getP
12  요청정보 전송방식   = <%=request.getM
13  요청 URL           = <%=request.getR
14  요청 URI           = <%=request.getR
15  컨텍스트 경로       = <%=request.getO
16  서버이름           = <%=request.getServerName() %><br>
17  서버 포트           = <%=request.getServerPort() %><br>
18 </body>
19 </html>
20
```



request 기본 객체 정보 실습

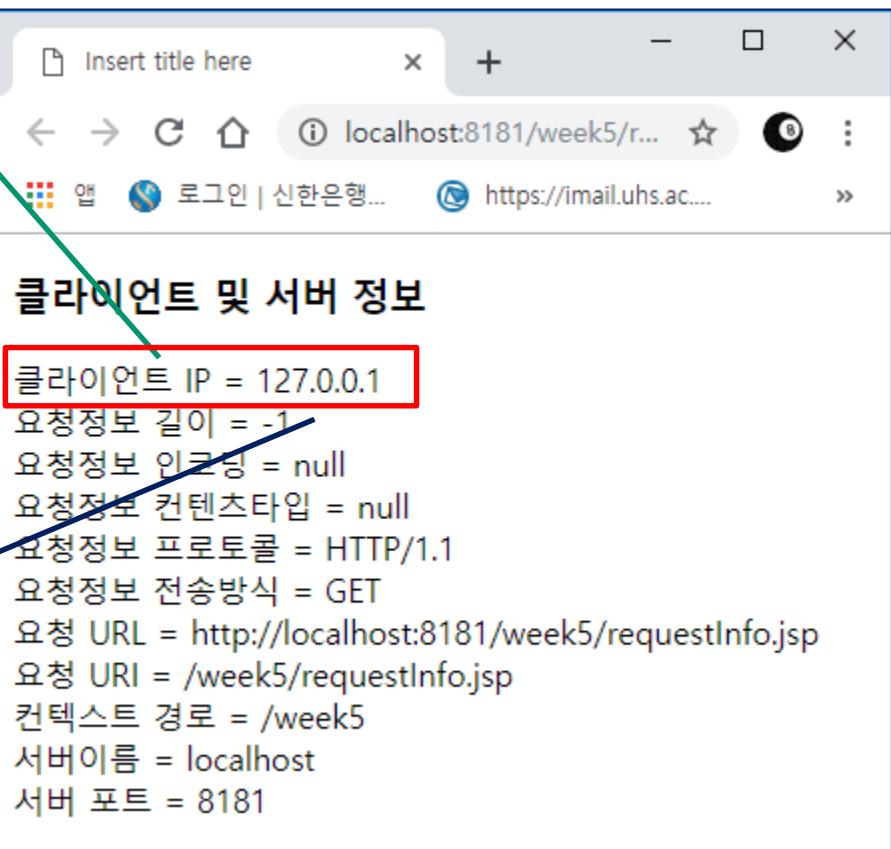
- requestInfo.jsp

클라이언트 IP = 0:0:0:0:0:0:1 => IPv6 방식

```
1 <%@ page language="java" contentType=
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html><html><head>
4 <meta charset="UTF-8"><title>Insert t
5 <body>
6     <h3>클라이언트 및 서버 정보</h3>
7     클라이언트 IP = <%
8         request.getR
9         request.getO
10        request.getO
11        request.getO
12        request.getO
13        request.getO
14        request.getO
15        request.getO
16        request.getO
17        request.getServerName() %><br>
18    </body>
19 </html>
20
```

IPv4 형식으로 바꾸는 방법

- eclipse > Run > Run configuration > Apache Tomcat > Tomcat 9.0 > Arguments 탭 선택
- VM arguments 에 다음 입력
- -Djava.net.preferIPv4Stack=true
- Apply 버튼 클릭
- **Tomcat 서버를 중지 후 다시 시작**



request 기본 객체

- 요청 파라미터 관련 메서드

메서드	리턴 타입	설명
<code>getParameter(String name)</code>	<code>String</code>	이름이 name인 파라미터의 값을 구한다. 존재하지 않을 경우 null을 리턴한다
<code>getParameterValues(String name)</code>	<code>String[]</code>	이름이 name인 모든 파라미터의 값을 배열로 구한다. 존재하지 않을 경우 null을 리턴한다
<code>getParameterNames()</code>	<code>java.util.Enumeration</code>	웹 브라우저가 전송한 파라미터의 이름 목록 을 구한다
<code>getParameterMap()</code>	<code>java.util.Map</code>	웹 브라우저가 전송한 파라미터의 맵(Map) 을 구한다. 맵은 <파라미터 이름, 값>의 쌍으로 구성된다

request 기본 객체 요청 파라미터 실습

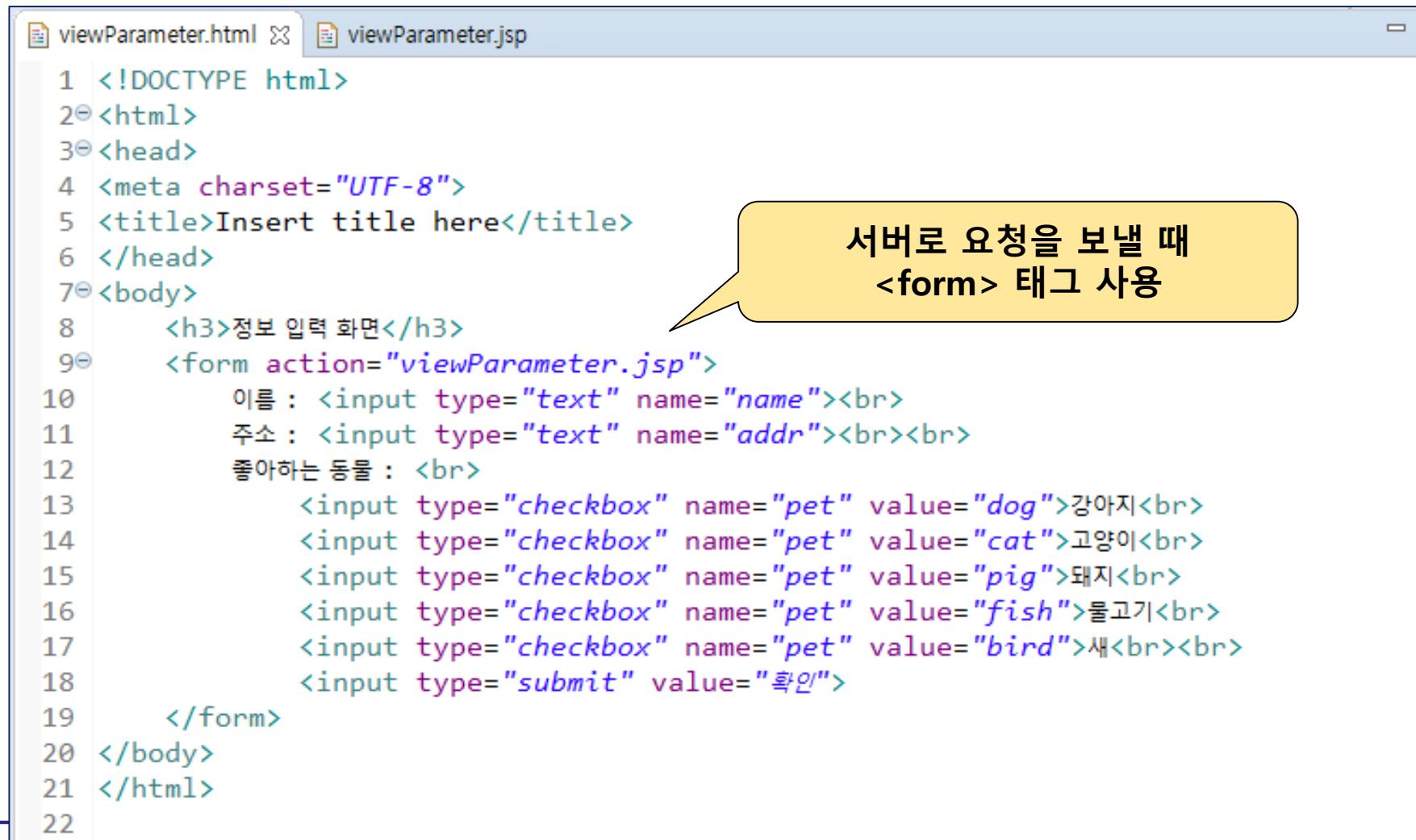
- 요청 파라미터 관련 메서드 실습(viewParameter.html)

The screenshot shows a web browser window with the following details:

- Title Bar:** Insert title here
- Address Bar:** localhost:8181/week5/... (with a star icon)
- Toolbar:** Back, Forward, Stop, Refresh, Home, Favorites, Help, and a search bar.
- Navigation Bar:** App, 로그인 | 신한은행..., https://imail.uhs.ac....
- Content Area:**
 - Section:** 정보 입력 화면
 - Text Input:** 이름 :
 - Text Input:** 주소 :
 - Section:** 좋아하는 동물 :
 - Checkboxes:**
 - 강아지
 - 고양이
 - 돼지
 - 물고기
 - 새
 - Button:** 확인

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(viewParameter.html)



viewParameter.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8     <h3>정보 입력 화면</h3>
9     <form action="viewParameter.jsp">
10         이름 : <input type="text" name="name"><br>
11         주소 : <input type="text" name="addr"><br><br>
12         좋아하는 동물 : <br>
13             <input type="checkbox" name="pet" value="dog">강아지<br>
14             <input type="checkbox" name="pet" value="cat">고양이<br>
15             <input type="checkbox" name="pet" value="pig">돼지<br>
16             <input type="checkbox" name="pet" value="fish">들고기<br>
17             <input type="checkbox" name="pet" value="bird">새<br><br>
18             <input type="submit" value="확인">
19     </form>
20 </body>
21 </html>
```

서버로 요청을 보낼 때
<form> 태그 사용

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(viewParameter.html)

The screenshot shows a Java IDE interface with two files open: `viewParameter.html` and `viewParameter.jsp`. The `viewParameter.html` file contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8     <h3>정보 입력 화면</h3>
9     <form action="viewParameter.jsp">
10        이름 : <input type="text" name="name"><br>
11        주소 : <input type="text" name="addr"><br><br>
12        좋아하는 동물 : <br>
13            <input type="checkbox" name="pet" value="dog">강아지<br>
14            <input type="checkbox" name="pet" value="cat">고양이<br>
15            <input type="checkbox" name="pet" value="pig">돼지<br>
16            <input type="checkbox" name="pet" value="fish">들고기<br>
17            <input type="checkbox" name="pet" value="bird">새<br><br>
18            <input type="submit" value="확인">
19     </form>
20 </body>
21 </html>
```

To the right, a browser window displays the form with the following fields:

정보 입력 화면

이름 :

주소 :

좋아하는 동물 :

강아지
 고양이
 돼지
 물고기
 새

request 요청 파라미터 실습(viewParameter.jsp)

getParameter()

한 개의 파라미터 값을 받을 때

```
<body>
    <h2>요청 파라미터 출력</h2>
    <h3 style="color: red;">파라미터 값 읽기 : request.getParameter() 메서드 사용</h3>
    name 파라미터 = <%=request.getParameter("name") %> <br>
    addr 파라미터 = <%=request.getParameter("addr") %> <br>

    <p>
        <h3 style="color: red;">파라미터 값 목록 읽기 : request.getParameterValues() 메서드 사용</h3>
        <%
            String[] values = request.getParameterValues("pet");
            if (values != null){
                for (String p : values)
                    out.println(p);
            }
        %>
```

getParameterValues()

여러 개의 파라미터 값을 받을 때

request 요청 파라미터 실습(viewParameter.jsp)

```
<p>
<h3 style="color: red;">파라미터 이름 목록 : request.getParameterNames() 메서드 사용</h3>
<%
Enumeration pNames = request.getParameterNames();
while(pNames.hasMoreElements()){
    String pName = (String)pNames.nextElement();
    out.println(pName);
}
%>

<p>
<h3 style="color: red;">맵 형식으로 파라미터 값 읽기 : request.getParameterMap() 메서드 사용</h3>
<%
Map pMap = request.getParameterMap();
String[] value = (String[])pMap.get("name");
out.println("Map 형식 : (name, " + value[0] + ")");
%>

</body>
```

getParameterNames()
파라미터의 이름 목록

getParameterMap()
<파라미터 이름, 값>의
쌍으로 구성

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(infoBasic.html)

The image shows two side-by-side browser windows. The left window displays the 'infoBasic.html' page with a form for inputting student information. The right window displays the 'subject.jsp' page showing the selected subjects.

Left Window (infoBasic.html):

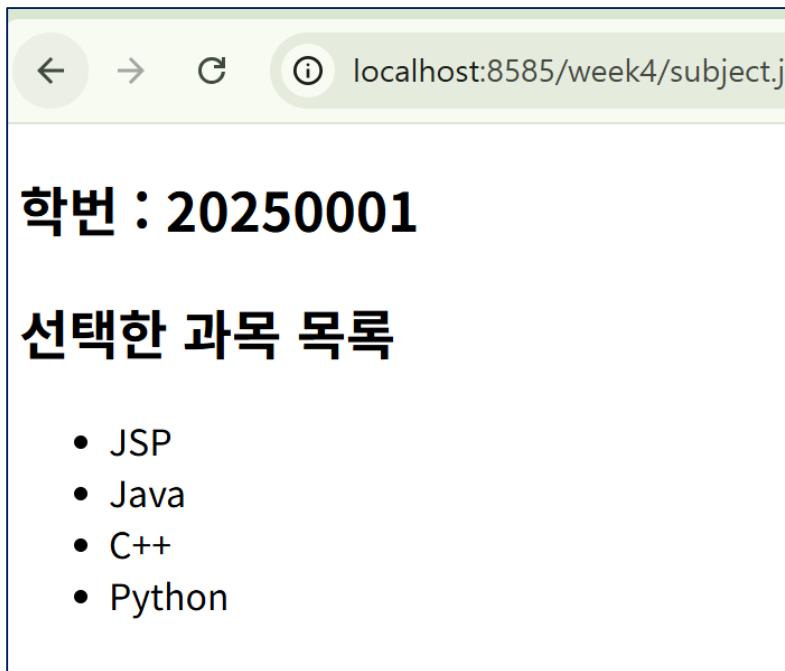
- URL: localhost:8585/week4/infoBasic.html
- Form Fields:
 - 학번: [Input Field]
 - 신청 과목:
 - JSP
 - Java
 - C++
 - Python
 -

Right Window (subject.jsp):

- URL: localhost:8585/week4/subject.jsp
- Output:
 - 학번 : 20250001
 - 선택한 과목 목록
 - JSP
 - Java
 - C++
 - Python

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(subject.jsp)



request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(infoBasic.html)

```
<body>
  <h2>수강 신청 정보 입력</h2>
  <form action="subject.jsp" method="get">
    학번: <input type="text" name="stnum"><br><br>
    신청 과목: <br>
    <input type="checkbox" name="subject" value="JSP"> JSP<br>
    <input type="checkbox" name="subject" value="Java"> Java<br>
    <input type="checkbox" name="subject" value="C++"> C++<br>
    <input type="checkbox" name="subject" value="Python"> Python<br><br>
    <input type="submit" value="전송">
  </form>

</body>
```

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(infoBasic.html)

```
<body>
<%
    String stNum = request.getParameter("stnum");
    String[] subjects = request.getParameterValues("subject");
%>

<h2> 학번 : <%=stNum %> </h2>
<h2> 선택한 과목 목록</h2>
<ul>
    <%
        if(subjects != null){
            for(String s : subjects){
                out.println("<li>" + s + "</li>");
            }
        } else
            out.println("<li>선택한 과목이 없습니다.</li>");
    %>
</ul>
</body>
```

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(infoBasic.html) - 추가사항

int age = Integer.parseInt(request.getParameter("age"));

The screenshot shows a Java web application running in a browser. On the left, a form titled "수강 신청 정보 입력" (Enrollment Information Input) is displayed. It contains fields for "학번" (Student ID) and "나이" (Age), both of which are highlighted with a red border. Below these are checkboxes for selecting subjects: JSP, Java, C++, and Python. A "전송" (Send) button is at the bottom. On the right, the processed output is shown in a box with a red border. It displays the student ID "학번 : 20250001" and the age "나이 : 25 (성인)". Below this, a section titled "선택한 과목 목록" (List of Selected Subjects) lists the checked subjects: JSP, Java, C++, and Python.

수강 신청 정보 입력

학번: [Red Box]

나이: [Red Box]

신청 과목:

JSP
 Java
 C++
 Python

전송

학번 : 20250001

나이 : 25 (성인)

선택한 과목 목록

- JSP
- Java
- C++
- Python

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(infoBasic.html)

```
<form action="subject.jsp" method="get">
    학번: <input type="text" name="stnum"><br>
    나이: <input type="text" name="age"><br><br>
    선성 과목: <br>
        <input type="checkbox" name="subject" value="JSP"> JSP<br>
        <input type="checkbox" name="subject" value="Java"> Java<br>
        <input type="checkbox" name="subject" value="C++"> C++<br>
        <input type="checkbox" name="subject" value="Python"> Python<br><br>
    <input type="submit" value="전송">
</form>
```

request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(subject.html)

```
<%
    String stNum = request.getParameter("stnum");
    int age = Integer.parseInt(request.getParameter("age"));
    String[] subjects = request.getParameterValues("subject");
%>

<h2> 학번 : <%=stNum %> </h2>
<h2> 나이 : <%=age %>
<%
    if(age >= 20)
        out.println(" (성인)");
    else
        out.println(" (미성년자)");
%>
</h2>
```

request 기본 객체

- **Header 관련 메서드**

메서드	리턴 타입	설명
getHeader(String name)	String	지정한 이름의 헤더 값을 구한다
getHeaders(String name)	java.util.Enumeration	지정한 이름의 헤더 목록을 구한다
getHeaderNames()	java.util.Enumeration	모든 헤더의 이름을 구한다
getIntHeader(String name)	int	지정한 헤더의 값을 정수 값으로 읽어온다
getDateHeader(String name)	long	지정한 헤더의 값을 시간 값으로 읽어온다

request Header 관련 메서드 실습(viewHeader.jsp)

viewHeaderList.jsp

```
1 <%@page import="java.util.Enumeration"%>
2 <%@ page language="java" contentType="text/html; charset=UTF-8"
3     pageEncoding="UTF-8"%>
4 <!DOCTYPE html>
5<html>
6<head>
7 <meta charset="UTF-8">
8 <title>Insert title here</title>
9 </head>
10<body>
11<h2>request 객체의 Header 정보</h2>
12<%
13     Enumeration headerEnum = request.getHeaderNames();
14     while(headerEnum.hasMoreElements()){
15         String hName = (String)headerEnum.nextElement();
16         String hValue = request.getHeader(hName);
17         out.println(hName + " = " + hValue + "<br>");
18     }
19     %>
20 </body>
21 </html>
22
```

request Header 관련 메서드 실습(viewHeader.jsp)

request 객체의 Header 정보

```
host = localhost:8585
connection = keep-alive
sec-ch-ua = "Not;A=Brand";v="99", "Google Chrome";v="139", "Chromium";v="139"
sec-ch-ua-mobile = ?0
sec-ch-ua-platform = "Windows"
upgrade-insecure-requests = 1
user-agent = Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0
Safari/537.36
accept =
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.7
sec-fetch-site = none
sec-fetch-mode = navigate
sec-fetch-user = ?1
sec-fetch-dest = document
accept-encoding = gzip, deflate, br, zstd
accept-language = ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
cookie = JSESSIONID=865A42CEC013CE2B93799F16E2D10D54
```

viewHeaderList.jsp

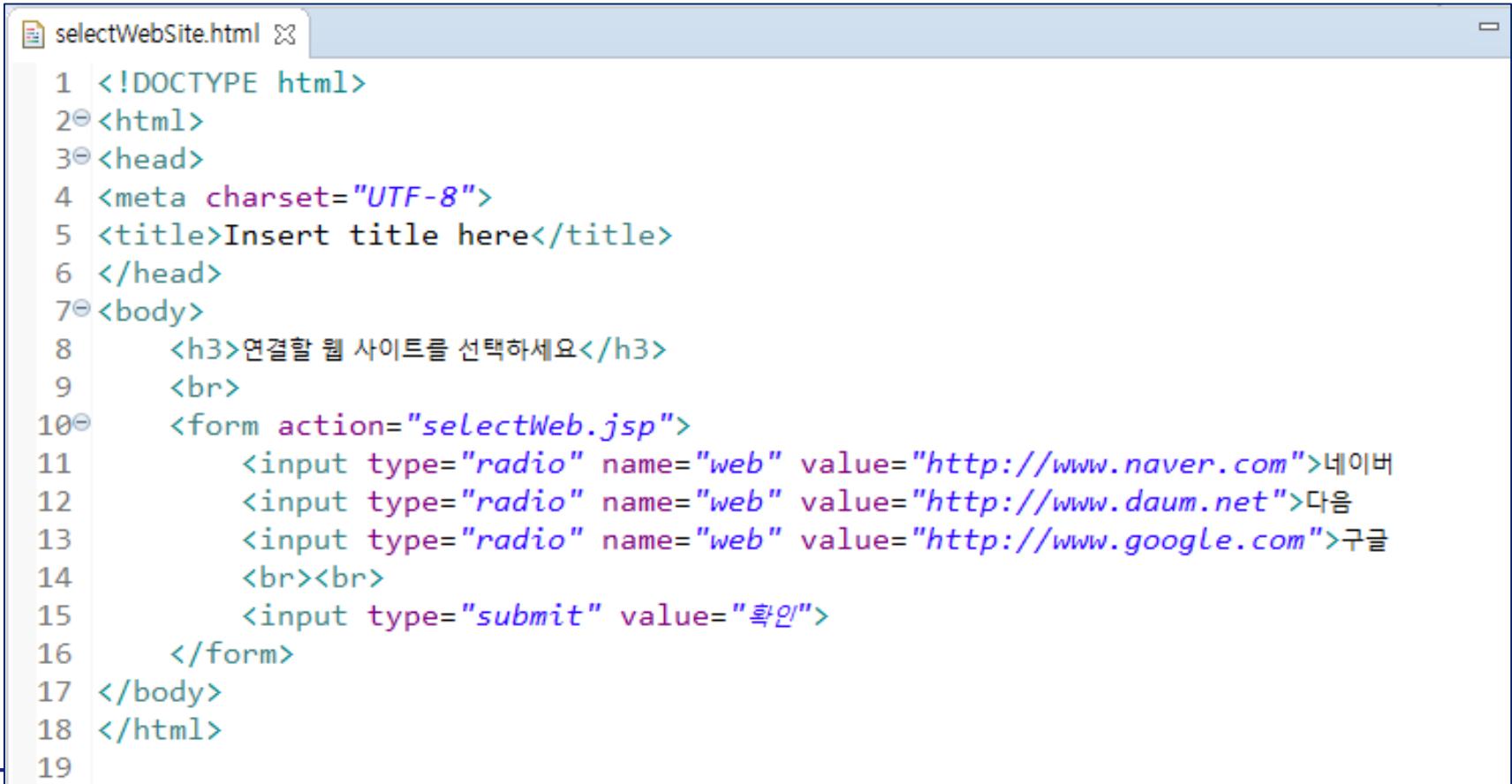
```
1 <%@page import="java.util.Enumeration,java.util.*"
2 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
3 <%@ page import="java.util.Enumeration,java.util.*"
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="UTF-8">
8 <title>Insert</title>
9 </head>
10 <body>
11 <h2>request 객체의 Header 정보</h2>
12 <% Enumeration headers = request.getHeaderNames(); %>
13 <% while(headers.hasMoreElements()) { %>
14 <% String headerName = headers.nextElement().toString();
15 <% String headerValue = request.getHeader(headerName);
16 <% System.out.println(headerName + " = " + headerValue); %>
17 <% } %>
18 <% %>
19 </body>
20 </html>
```

response 기본 객체

- 속성
 - request 기본 객체와 반대의 기능을 수행
 - request 기본 객체가 웹 브라우저가 전송한 요청 정보를 담고 있다면 response 기본 객체는 웹 브라우저에 보내는 응답 정보를 담는다
- 기능
 - 헤더 정보 입력
 - 리다이렉트(redirect) 기능

response 기본 객체

- **redirect** 기능을 이용해서 페이지 이동 가능
- redirect는 웹 서버가 웹 브라우저에게 다른 페이지로 이동하라고 응답하는 기능(selectWebSite.html)



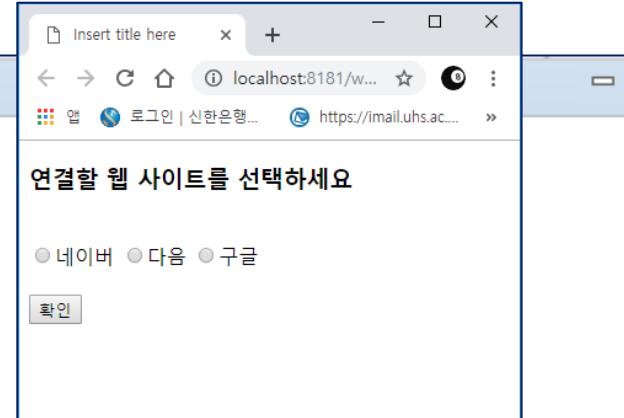
The screenshot shows a code editor window with the title bar "selectWebSite.html". The main area contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8     <h3>연결할 웹 사이트를 선택하세요</h3>
9     <br>
10    <form action="selectWeb.jsp">
11        <input type="radio" name="web" value="http://www.naver.com">네이버
12        <input type="radio" name="web" value="http://www.daum.net">다음
13        <input type="radio" name="web" value="http://www.google.com">구글
14        <br><br>
15        <input type="submit" value="확인">
16    </form>
17 </body>
18 </html>
19
```

response 기본 객체

- redirect 기능을 이용해서 페이지 이동 가능
- redirect는 웹 서버가 웹 브라우저에게 다른 페이지로 이동하라고 응답하는 기능 (selectWebSite.html)

```
1 <!DOCTYPE html>
2<html>
3<head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7<body>
8     <h3>연결할 웹 사이트를 선택하세요</h3>
9     <br>
10    <form action="selectWeb.jsp">
11        <input type="radio" name="web" value="http://www.naver.com">네이버
12        <input type="radio" name="web" value="http://www.daum.net">다음
13        <input type="radio" name="web" value="http://www.google.com">구글
14        <br><br>
15        <input type="submit" value="확인">
16    </form>
17 </body>
18 </html>
19
```



The screenshot shows a web browser window with the URL `localhost:8181/w...`. A modal dialog box is displayed, asking "연결할 웹 사이트를 선택하세요". It contains three radio buttons: "네이버" (Naver), "다음" (Daum), and "구글" (Google). Below the radio buttons is a "확인" (Confirm) button.

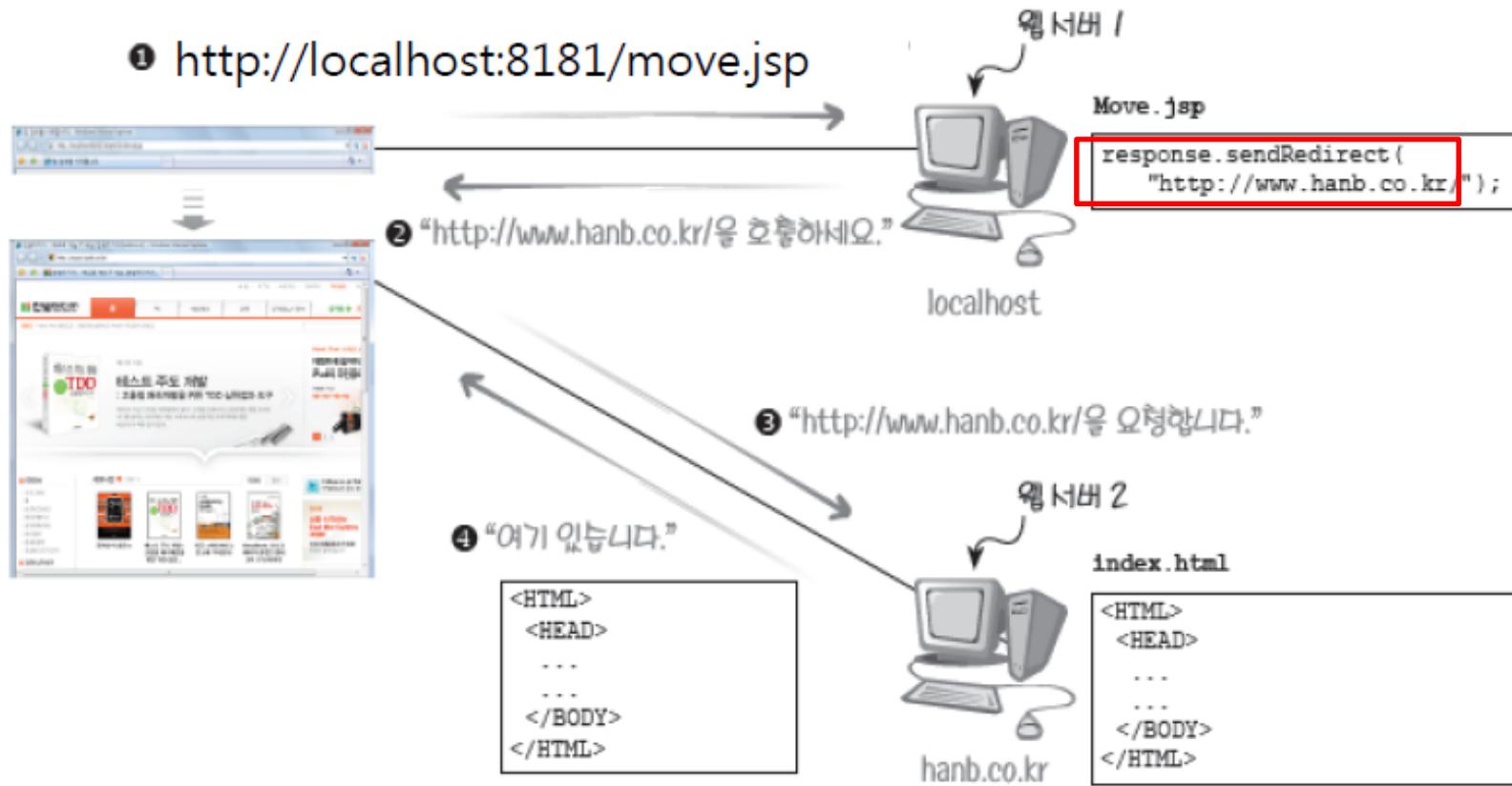
response 기본 객체

- redirect 기능을 이용해서 페이지 이동 가능
- redirect는 웹 서버가 웹 브라우저에게 다른 페이지로 이동하라고 응답하는 기능 (selectWeb.jsp)

```
selectWeb.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <%
11     String web = request.getParameter("web");
12     response.sendRedirect(web);
13 <%
14 </body>
15 </html>
16
```

response 기본 객체

- **sendRedirect** 메서드는 파라미터로 지정된 URL을 직접 호출하는 것이 아니라 그 URL을 이용해서 다시 웹 자원을 호출하라는 메시지를 웹 브라우저로 보내기만 한다



과제물

- 개인 정보 입력 및 결과 화면 만들기

< 개인정보 입력 화면 >

1. 이름 :

2. 나이 :

3. 성별 : 여자 남자

4. 주소 :

5. 좋아하는 운동

농구

축구

탁구

헬스

입력

< 개인정보 입력 결과 >

구분	내용
이름	홍길동
나이	25세
성별	남자
주소	인천시 미추홀구 xxx동
좋아하는 운동	탁구, 헬스

과제물

- 제출방법
 - 프로젝트명(Dynamic Web Project) : **hw2_학번**
 - homework2.html
 - homework2.jsp
 - 프로젝트명 오른쪽 마우스 클릭 > **Export** > **War file** 선택
 - Web project : 학번
 - Destination : 저장하고자 하는 폴더 선택
 - 최종 생성된 **hw2_학번.war** 파일을 제출

과제물

- 개인 정보 입력 및 결과 화면 만들기

- 입력 화면과 출력 화면 이미지로 저장해서 제출
- 개인 정보는 본인 이름, 나이, 성별, 간략 주소 입력

< 개인정보 입력 화면 >

- 이름 :
- 나이 :
- 성별 : 여자 남자
- 주소 :
- 좋아하는 운동
 - 농구
 - 축구
 - 탁구
 - 헬스

입력

< 개인정보 입력 결과 >

구분	내용
이름	홍길동
나이	25세
성별	남자
주소	인천시 미추홀구 xxx동
좋아하는 운동	탁구, 헬스