



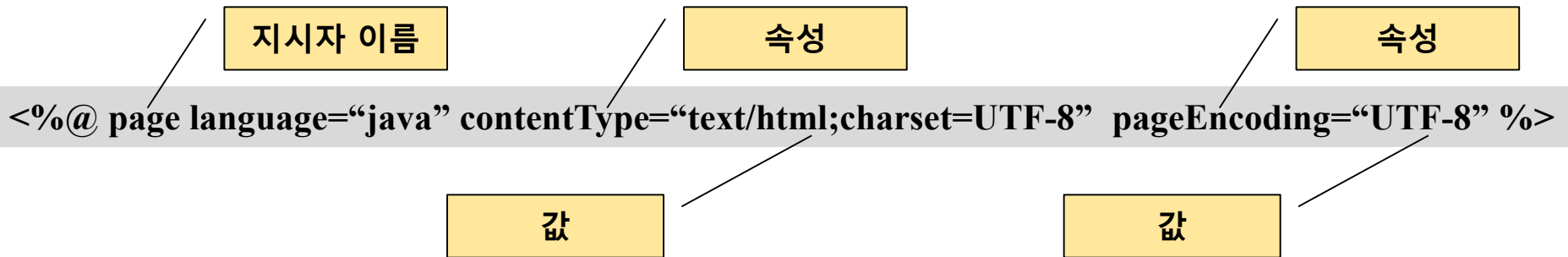
## JSP 구성 요소

# JSP 구성 요소

- 디렉티브(Directive) – 지시자
- 스크립트
  - 스크립틀릿(scriptlet)
  - 표현식(expression)
  - 선언부(declaration)
- 기본 객체
- 표준 액션 태그
- 표현언어(Expression Language)
- 커스텀 태그와 표준 태그 라이브러리(JSTL)

# 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
  - **taglib** : 태그 라이브러리를 지정
  - **include** : 특정 영역에 다른 문서를 포함

# 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
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

- info 속성 – JSP 페이지에 대한 설명

## < pageInfo.jsp >

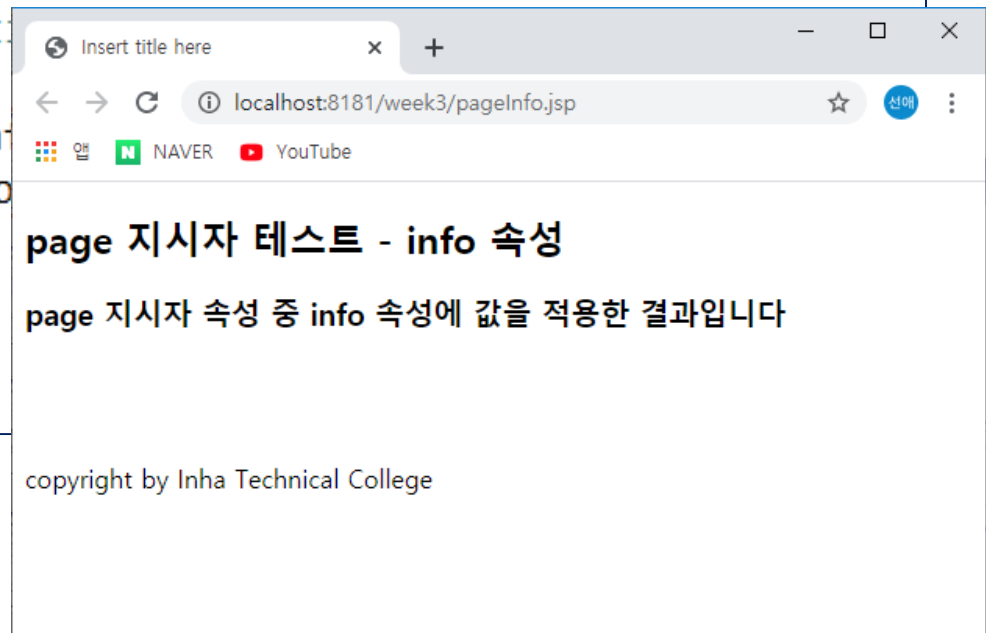
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page info="copyright by Inha Technical College" %>
4 <!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6 <body>
7   <h2>page 지시자 테스트 - info 속성</h2>
8   <h3>page 지시자 속성 중 info 속성에 값을 적용한 결과입니다</h3>
9   <br><br><br>
10  <%=getServletInfo() %>
11 </body>
12 </html>
```

# page 지시자 속성 실습

- info 속성 – JSP 페이지에 대한 설명

## < pageInfo.jsp >

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page info="copyright by Inha Technical College" %>
4 <!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title>
6 <body>
7   <h2>page 지시자 테스트 - info 속성
8   <h3>page 지시자 속성 중 info 속성
9   <br><br><br>
10  <%=getServletInfo() %>
11 </body>
12 </html>
```



# page 지시자 속성 실습

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

< dateTime.jsp >

```
1 <%@page import="java.util.Calendar"%>
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"%>
6 <!DOCTYPE html><html><head><meta charset="UTF-8">
7 <title>Insert title here</title></head>
8 <body>
9 <%
10     Date date = new Date();
11     SimpleDateFormat sdf = new SimpleDateFormat("hh:mm:ss");
12     String time = sdf.format(date);
13     Calendar cal = Calendar.getInstance();
14 %>
15 <h1>오늘은 <%= cal.get(Calendar.YEAR) %>년
16             <%= cal.get(Calendar.MONTH)+1 %>월
17             <%= cal.get(Calendar.DATE) %>일입니다</h1>
18     <h2>현재 시각 <%= time %></h2>
19 </body></html>
```

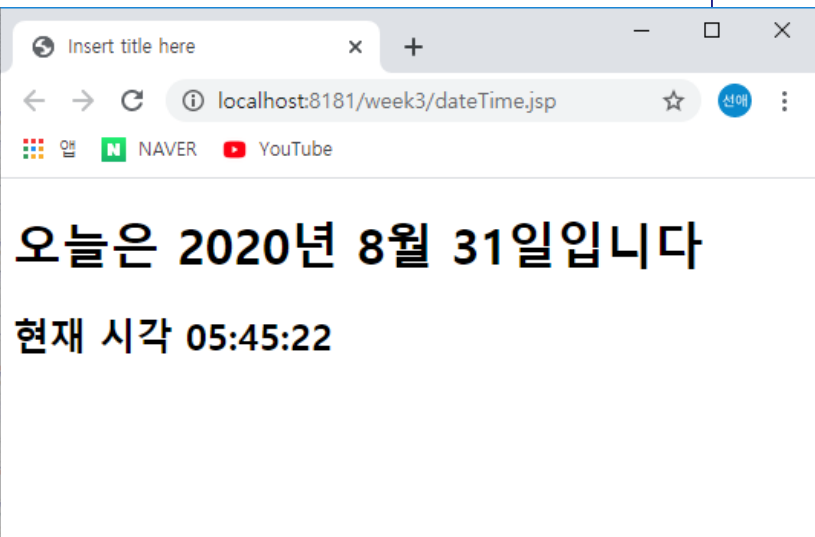


# page 지시자 속성 실습

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

< dateTime.jsp >

```
1 <%@page import="java.util.Calendar"%>
2 <%@page import="java.text.SimpleDateFormat"%>
3 <%@page import="java.util.Date"%>
4 <%@ page language="java" contentType="text/html"
5   pageEncoding="UTF-8"%>
6 <!DOCTYPE html><html><head><meta
7 <title>Insert title here</title><
8 <body>
9 <%
10     Date date = new Date();
11     SimpleDateFormat sdf = new Si
12     String time = sdf.format(date
13     Calendar cal = Calendar.getIn
14 %>
15 <h1>오늘은 <%= cal.get(Calendar.YEAR) %>년
16           <%= cal.get(Calendar.MONTH)+1 %>월
17           <%= cal.get(Calendar.DATE) %>일입니다</h1>
18 <h2>현재 시각 <%= time %></h2>
19 </body></html>
```



# include 지시자

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

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

현재 디렉터리에 있는 dateTime.jsp 를 include 한다

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

하위 sub1 디렉터리에 있는 dateTime.jsp 를 include 한다

# 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>
```

localhost:8181/week3/include.jsp

앱

## 가장 여행하고 싶은 곳은?

1. 프랑스
2. 헝가리
3. 이탈리아
4. 캐나다
5. 호주

오늘은 2019년 9월 4일입니다

현재 시각 05:39:48

# include 지시자 실습

- include 지시자의 사용 예

< include.jsp 페이지 소스 보기 결과 >

```
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>
```

← → ↻ ⓘ view-source:localhost:8181/week3/include.jsp

앱

```
1 <!DOCTYPE html><html><head><meta charset="UTF-8">
2 <title>Insert title here</title></head>
3 <body>
4 <h1>가장 여행하고 싶은 곳은?</h1>
5 <h3>1. 프랑스</h3>
6 <h3>2. 헝가리</h3>
7 <h3>3. 이탈리아</h3>
8 <h3>4. 캐나다</h3>
9 <h3>5. 호주</h3>
10
11
12
13
14
15 <!DOCTYPE html><html><head><meta charset="UTF-8">
16 <title>Insert title here</title></head>
17 <body>
18
19   <h1>오늘은 2019년
20       9월
21       4일입니다</h1>
22   <h2>현재 시각 05:39:48</h2>
23 </body></html>
24
25
26 </body></html>
```

# JSP 구성 요소

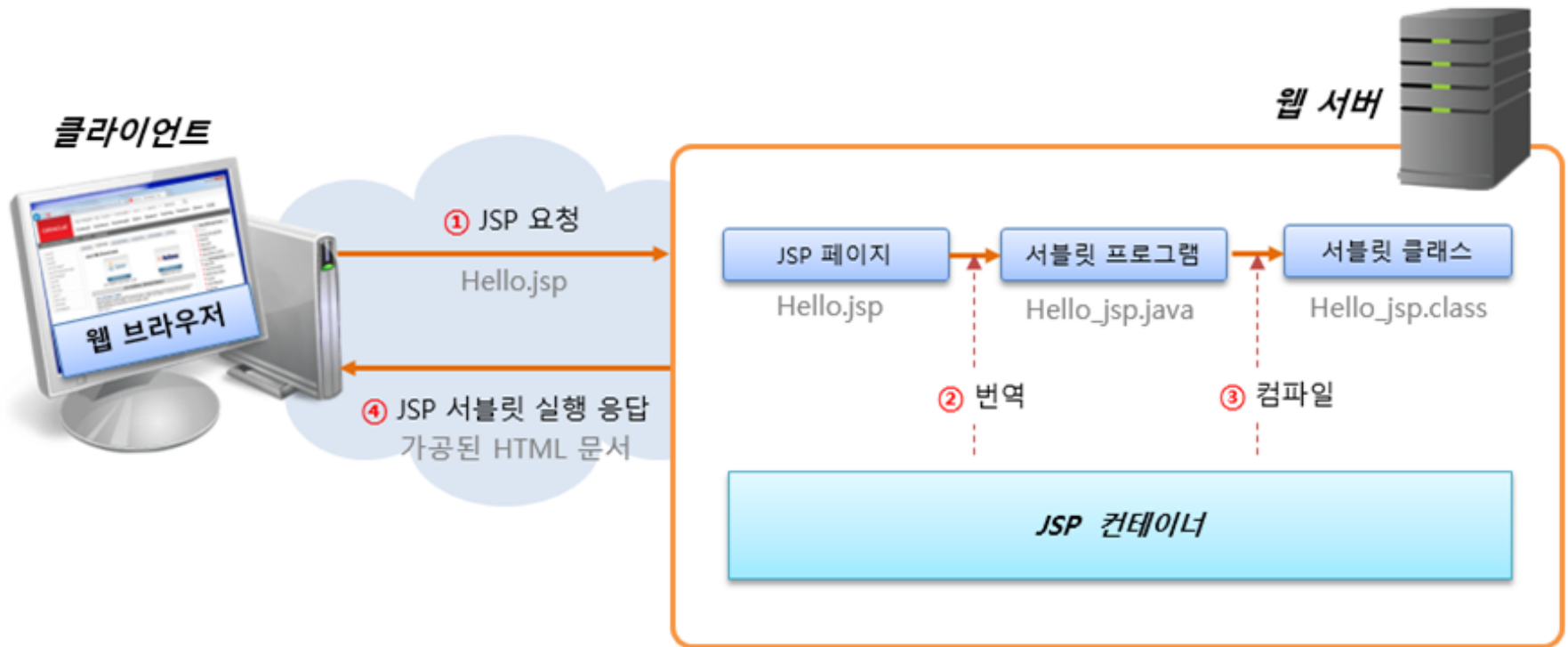
- 디렉티브(Directive) – 지시자
- 스크립트
  - 스크립틀릿(scriptlet)
  - 표현식(expression)
  - 선언부(declaration)
- 기본 객체
- 표준 액션 태그
- 표현언어(Expression Language)
- 커스텀 태그와 표준 태그 라이브러리(JSTL)

# 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

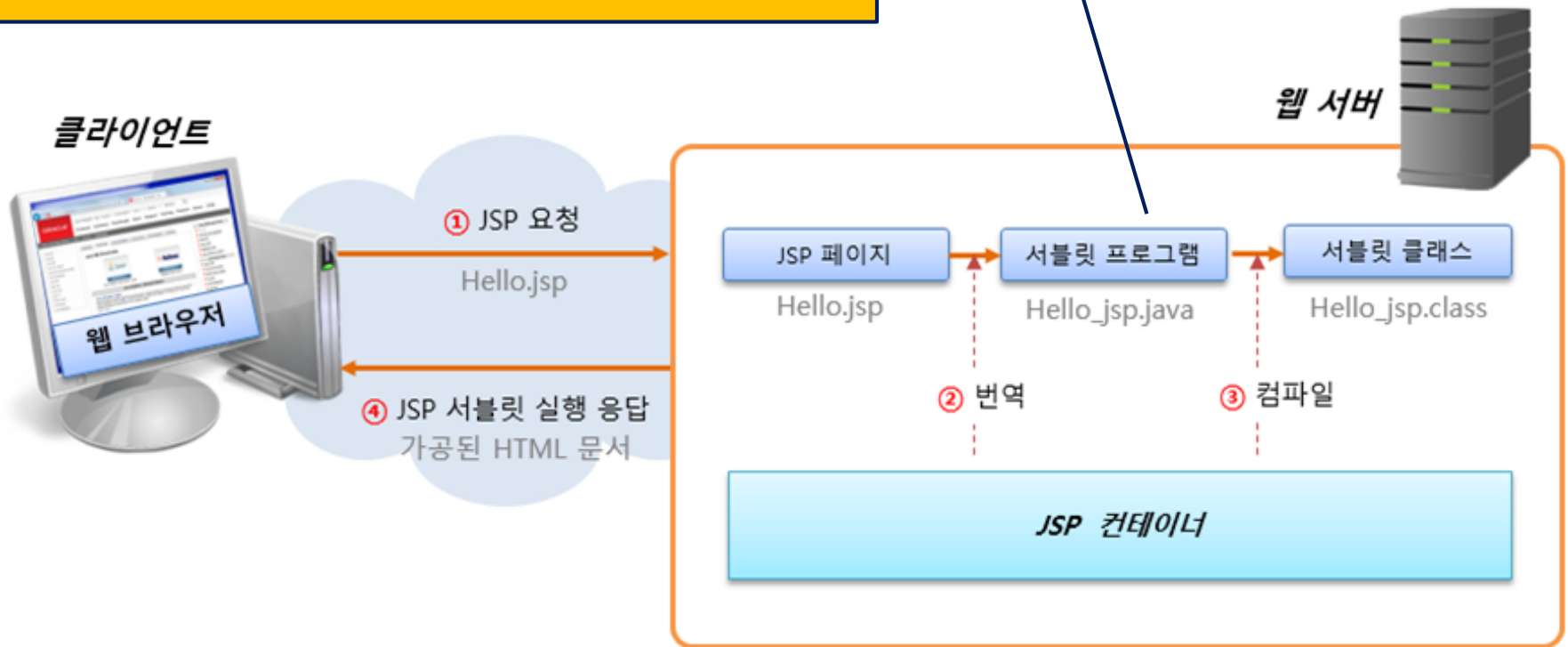
- 스크립틀릿(scriptlet)
- 표현식(expression)
- 선언문(declaration)





# 스크립트 구성 요소

생성된 자바 소스 코드 & 서블릿 클래스 위치 :  
workspace\metadata\plugins\org.eclipse.wst.server.core\wtp0\work\Catalina\localhost\week3\work\apache\jsp



# 스크립트 변수의 범위

- 스크립틀릿(scriptlet)에서 선언된 변수
  - 지역변수에 해당된다.
  - 별도로 선언된 메소드 부분에서는 해당 변수를 사용할 수 없다.
  - 자동 초기화가 안되기 때문에 반드시 초기화를 해 주어야 한다.
- 선언문(declaration)에서 선언된 변수나 메소드
  - 전역변수에 해당된다.
  - JSP 페이지 내 어디에서든지 사용할 수 있다.
  - 자동 초기화가 된다.

# 스크립트 변수의 범위 실습(varRange.jsp)

varRange.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   <h3>변수의 범위</h3>
7   <%!
8     //선언문 - 선언문 내에서 정의된 변수는 전역변수이다
9     String strVar = "전역변수입니다";
10  %>
11  <%!
12    //선언문 - 메소드 선언
13    String getStr() {
14      return strVar;
15    }
16  %>
17  <%
18    //스크립틀릿 - 스크립틀릿 내에서 정의된 변수는 지역변수이다
19    String strVar2 = "지역변수입니다";
20  %>
21  <br>
22  선언문에서 선언한 변수 strVar = <%=getStr() %><br>
23  스크립틀릿에서 선언한 변수 strVar2 = <%=strVar2 %><br>
24 </body>
25 </html>
```

# 스크립트 변수의 범위 실습

varRange.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   <h3>변수의 범위</h3>
7   <%!
8     //선언문 - 선언문 내에서 정의된 변수는 전역변수입니다
9     String strVar = "전역변수입니다"
10  %>
11   <%!
12     //선언문 - 메소드 선언
13     String getStr() {
14       return strVar;
15     }
16  %>
17   <%
18     //스크립틀릿 - 스크립틀릿 내에서 정의된 변수는 지역변수이다
19     String strVar2 = "지역변수입니다";
20  %>
21   <br>
22   선언문에서 선언한 변수 strVar = <%=getStr() %><br>
23   스크립틀릿에서 선언한 변수 strVar2 = <%=strVar2 %><br>
24 </body>
25 </html>
```

Insert title here

localhost:8181/...  
로그인 | 신한은행 ...  
https://imail.uhs.ac....

## 변수의 범위

선언문에서 선언한 변수 strVar = 전역변수입니다  
스크립틀릿에서 선언한 변수 strVar2 = 지역변수입니다

# 스크립트 변수의 범위 실습

```
varRange.jsp  varRange2.jsp  ✕  
9<body>  
10  <h3>변수의 범위</h3>  
11  <%!  
12      //선언문 - 선언문 내에서 정의된 변수는 전역변수이다  
13      /* String strVar = "전역변수입니다"; */  
14      String strVar;  
15  %>  
16  <%!  
17      //선언문 - 메소드 선언  
18      String getStr() {  
19          return strVar;  
20      }  
21  %>  
22  <%  
23      //스크립틀릿 - 스크립틀릿 내에서 정의된 변수는 지역변수이다  
24      /* String strVar2 = "지역변수입니다"; */  
25      String strVar2;  
26  %>  
27  <%!  
28      String getStr2(){  
29          return strVar2;  
30      }  
31  %>  
32  <br>  
33  선언문에서 선언한 변수 strVar = <%=getStr() %><br>  
34  스크립틀릿에서 선언한 변수 strVar2 = <%=strVar2 %><br>  
35  </body></html>
```

선언문 내에서 정의된  
변수에 초기값을 주지 않음

스크립틀릿 내에서 정의된  
변수에 초기값을 주지 않음

스크립틀릿 내에서 정의된  
변수를 메소드에서 사용함

# 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**
- 표준 액션 태그
- 표현언어(Expression Language)
- 커스텀 태그와 표준 태그 라이브러리(JSTL)



# 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 <b>요청 정보</b> 를 저장한다
response	javax.servlet.http.HttpServletResponse	<b>응답 정보</b> 를 저장한다
pageContext	javax.servlet.jsp.PageContext	JSP 페이지에 대한 정보를 저장한다
session	javax.servlet.http.HttpSession	<b>HTTP 세션 정보</b> 를 저장한다
application	javax.servlet.ServletContext	웹 어플리케이션에 대한 정보를 저장한다
out	javax.servlet.jsp.JspWriter	JSP 페이지가 생성하는 결과를 <b>출력할 때 사용하는</b> 출력 스트림이다
config	javax.servlet.ServletConfig	JSP 페이지에 대한 설정 정보를 저장한다
page	java.lang.Object	JSP 페이지를 구현한 자바 클래스 인스턴스이다
exception	java.lang.Throwable	익셉션 객체이다. <b>에러 페이지에서만</b> 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 주소를 구한다
getContentLength()	long	클라이언트가 전송한 요청 정보의 길이를 구한다
getCharacterEncoding()	String	클라이언트가 요청 정보를 전송할 때 사용한 캐릭터의 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
getRequestURI()	String	웹 브라우저가 요청한 URL에서 경로를 구한다
getContextPath()	String	JSP 페이지가 속한 웹 어플리케이션의 컨텍스트 경로를 구한다
getServerName()	String	연결할 때 사용한 서버 이름을 구한다
getServerPort()	int	서버가 실행 중인 포트 번호를 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 콘텐츠타입 = <%=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.getC
9     요청정보 인코딩     = <%=request.getC
10    요청정보 컨텐츠타입 = <%=request.getC
11    요청정보 프로토콜   = <%=request.getP
12    요청정보 전송방식    = <%=request.getM
13    요청 URL            = <%=request.getR
14    요청 URI             = <%=request.getR
15    컨텍스트 경로       = <%=request.getC
16    서버이름            = <%=request.getServerName() %><br>
17    서버 포트           = <%=request.getServerPort() %><br>
18 </body>
19 </html>
20
```

Insert title here

localhost:8181/week5/r...

로그인 | 신한은행... https://imail.uhs.ac...

## 클라이언트 및 서버 정보

클라이언트 IP = 0:0:0:0:0:0:1

요청정보 길이 = -1

요청정보 인코딩 = null

요청정보 컨텐츠타입 = null

요청정보 프로토콜 = HTTP/1.1

요청정보 전송방식 = GET

요청 URL = http://localhost:8181/week3/requestInfo.jsp

요청 URI = /week3/requestInfo.jsp

컨텍스트 경로 = /week3

서버이름 = localhost

서버 포트 = 8181

# request 기본 객체 정보 실습

- requestInfo.jsp

클라이언트 IP = 0:0:0:0:0:0:1 => IPv6 방식

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**

클라이언트 및 서버 정보

클라이언트 IP = 127.0.0.1

요청정보 길이 = -1  
요청정보 인코딩 = null  
요청정보 컨텐츠타입 = null  
요청정보 프로토콜 = HTTP/1.1  
요청정보 전송방식 = GET  
요청 URL = http://localhost:8181/week5/requestInfo.jsp  
요청 URI = /week5/requestInfo.jsp  
컨텍스트 경로 = /week5  
서버이름 = localhost  
서버 포트 = 8181



# request 기본 객체

- 요청 파라미터 관련 메서드

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

# request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(viewParameter.html)

```
viewParameter.html viewParameter.jsp
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">물고기<br>
17             <input type="checkbox" name="pet" value="bird">새<br><br>
18             <input type="submit" value="확인">
19     </form>
20 </body>
21 </html>
22
```

# request 기본 객체 요청 파라미터 실습

- 요청 파라미터 관련 메서드 실습(viewParameter.html)

viewParameter.html

viewParameter.jsp

```
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">물고기<br>
17             <input type="checkbox" name="pet" value="bird">새<br><br>
18             <input type="submit" value="확인">
19     </form>
20 </body>
21 </html>
22
```

Insert title here

localhost:8181/week5/...

로그인 | 신한은행... https://mail.uhs.ac...

정보 입력 화면

이름 :

주소 :

좋아하는 동물 :

☐ 강아지

☐ 고양이

☐ 돼지

☐ 물고기

☐ 새

확인

# request 요청 파라미터 실습(viewParameter.jsp)

```
viewParameter.html viewParameter.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@page import="java.util.Enumeration, java.util.Map"%>
4 <!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6 <body>
7   <% request.setCharacterEncoding("utf-8"); %>
8   <h3>요청 파라미터 출력</h3>
9   <h4 style="color: red">request.getParameter() 메서드 사용</h4>
10  name 파라미터 = <%=request.getParameter("name") %>,
11  addr 파라미터 = <%=request.getParameter("addr") %><p>
12
13  <h4 style="color: red">request.getParameterValues() 메서드 사용</h4>
14  <%
15    String[] values = request.getParameterValues("pet");
16    if (values != null){
17      for (int i=0; i<values.length; i++)
18        out.println(values[i]);
19    }
20  %> <p>
21  <h4 style="color: red">request.getParameterNames() 메서드 사용</h4>
22  <%
23    Enumeration paramEnum = request.getParameterNames();
24    while(paramEnum.hasMoreElements()){
25      String name = (String)paramEnum.nextElement();
26      out.println(name);
27    }
28  %> <p>
29  <h4 style="color: red">request.getParameterMap() 메서드 사용</h4>
30  <%
31    Map paramMap = request.getParameterMap();
32    String[] nameParam = (String[])paramMap.get("pet");
33    if(nameParam != null)
34      out.println(nameParam[1]);
35  %>
36 </body></html>
```

# request 요청 파라미터 실습(viewParameter.jsp)

```
viewParameter.html viewParameter.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@page import="java.util.Enumeration, java.util.Map"%>
4 <!DOCTYPE html><html><head><meta charset="UTF-8">
5 <title>Insert title here</title></head>
6 <body>
7   <% request.setCharacterEncoding("utf-8"); %>
8   <h3>요청 파라미터 출력</h3>
9   <h4 style="color: red">request.getParameter() 메서드 사용
10   name 파라미터 = <%=request.getParameter("name") %>,
11   addr 파라미터 = <%=request.getParameter("addr") %><p>
12
13   <h4 style="color: red">request.getParameterValues()
14   <%
15       String[] values = request.getParameterValues("
16       if (values != null){
17         for (int i=0; i<values.length; i++)
18           out.println(values[i]);
19       }
20   %> <p>
21   <h4 style="color: red">request.getParameterNames()
22   <%
23   Enumeration paramEnum = request.getParameterNames();
24   while(paramEnum.hasMoreElements()){
25     String name = (String)paramEnum.nextElement();
26     out.println(name);
27   }
28   %> <p>
29   <h4 style="color: red">request.getParameterMap() 메서드 사용</h4>
30   <%
31   Map paramMap = request.getParameterMap();
32   String[] nameParam = (String[])paramMap.get("pet");
33   if(nameParam != null)
34     out.println(nameParam[1]);
35   %>
36 </body></html>
```

Insert title here x + - □ ×

← → ↺ ⌂ ⓘ localhost:8181/week5/... ☆ 🌙 ⋮

앱 로그인 | 신한은행... https://imail.uhs.ac.... »

### 요청 파라미터 출력

**request.getParameter() 메서드 사용**

name 파라미터 = 홍길동, addr 파라미터 = 서울

**request.getParameterValues() 메서드 사용**

dog cat fish bird

**request.getParameterNames() 메서드 사용**

name addr pet

**request.getParameterMap() 메서드 사용**

cat

# 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 <%
12     Enumeration headerEnum = request.getHeaderNames();
13     while (headerEnum.hasMoreElements()){
14         String hName = (String)headerEnum.nextElement();
15         String hValue = request.getHeader(hName);
16         out.println(hName + " = " + hValue + "<br>");
17     }
18 %>
19 </body>
20 </html>
21
22
```

# request Header 관련 메서드 실습(viewHeader.jsp)

viewHeaderList.jsp

```
1 <%@page import="java.util.Enumeration"%>
2 <%@ page language="java" %>
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     <%
12         Enumeration headerEnum = request.getHeaderNames();
13         while (headerEnum.hasMoreElements()) {
14             String hName = headerEnum.nextElement();
15             String hValue = request.getHeader(hName);
16             out.println(hName + " = " + hValue + "<br>");
17         }
18     %>
19 </body>
20 </html>
21
22
```

Insert title here

localhost:8181/week5/viewHeaderList.jsp

앱 로그인 | 신한은행... https://iemail.uhs.ac... 협성대학교 도서관 Flaticon - Freepik

host = localhost:8181  
connection = keep-alive  
upgrade-insecure-requests = 1  
user-agent = Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36  
accept = text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3  
accept-encoding = gzip, deflate, br  
accept-language = ko,en-US;q=0.9,en;q=0.8  
cookie = JSESSIONID=6AF6ED5C9E82A5BA99F1FCA0B6734675



# response 기본 객체

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

# response 기본 객체

- 응답 정보에 헤더를 추가하는 기능을 제공

메서드	설명
<code>addDateHeader(String name, long date)</code>	name 헤더에 date를 추가한다.
<code>addHeader(String name, String value)</code>	name 헤더에 value를 추가한다
<code>addIntHeader(String name, int value)</code>	name 헤더에 정수 값 value를 추가한다
<code>setDateHeader(String name, long date)</code>	name 헤더의 값을 date로 지정한다
<code>setHeader(String name, String value)</code>	name 헤더의 값을 value로 지정한다
<code>setIntHeader(String name, int value)</code>	name 헤더의 값을 정수 값 value로 지정한다
<code>containsHeader(String name)</code>	이름이 name인 헤더를 포함하고 있으면 true를, 그렇지 않으면 false 리턴

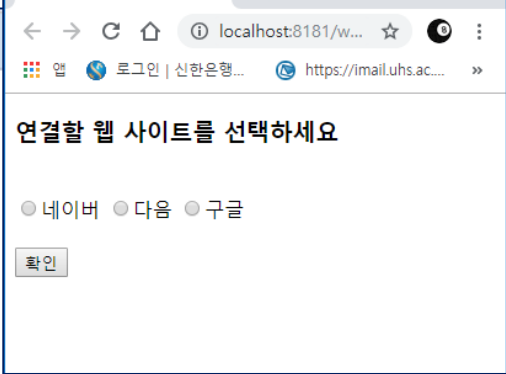
# response 기본 객체

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

```
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
```

# 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
```

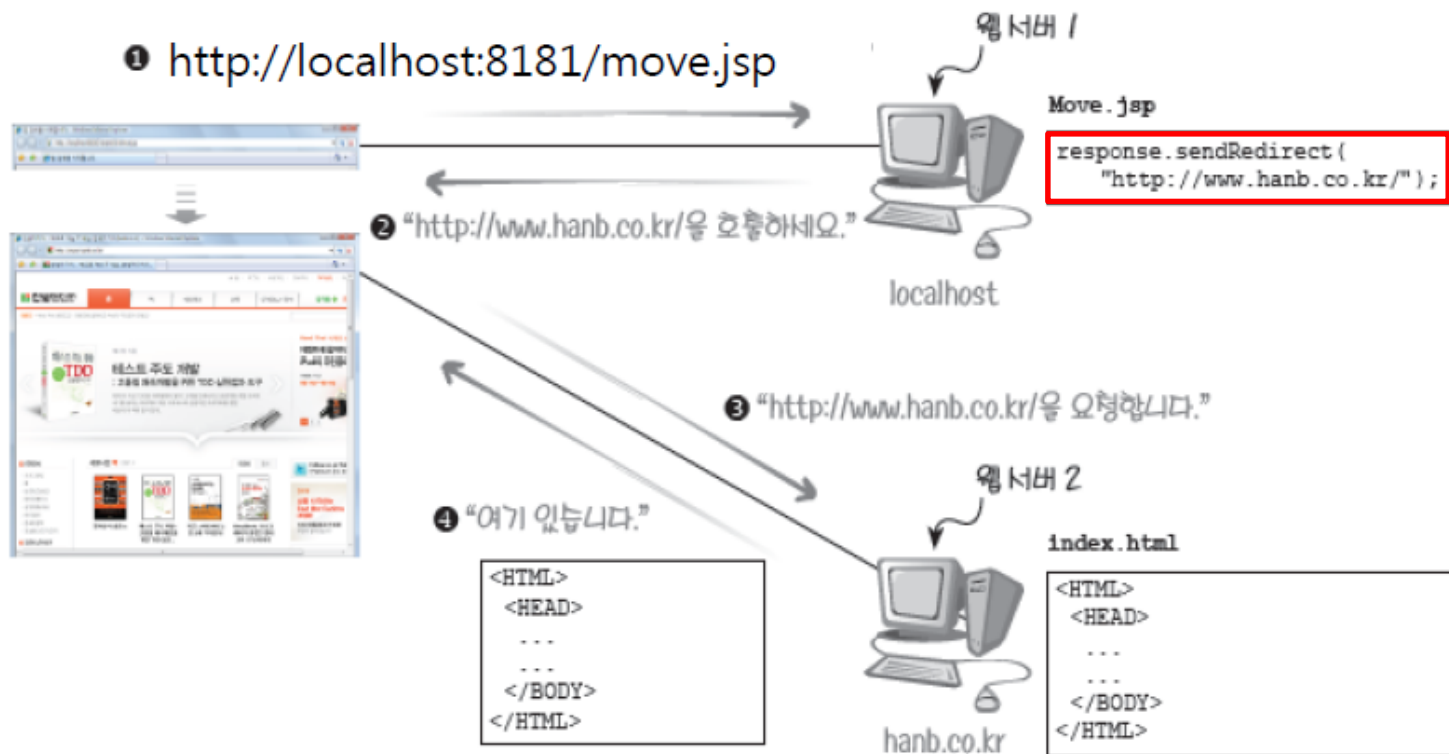
# response 기본 객체

- 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



sendRedirect 메서드의 작동 원리

## 과제 #2

- 개인 정보 입력 및 결과 화면 만들기

### < 개인정보 입력 화면 >

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

입력

### < 개인정보 입력 결과 >

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

# 과제 #2

- 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



## 과제 #2

- 개인 정보 입력 및 결과 화면 만들기

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

### < 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	내 용
이름	홍길동
나이	25세
성별	남자
주소	인천시 미추홀구 xxx동
좋아하는 운동	탁구, 헬스