



CHAPTER 15.

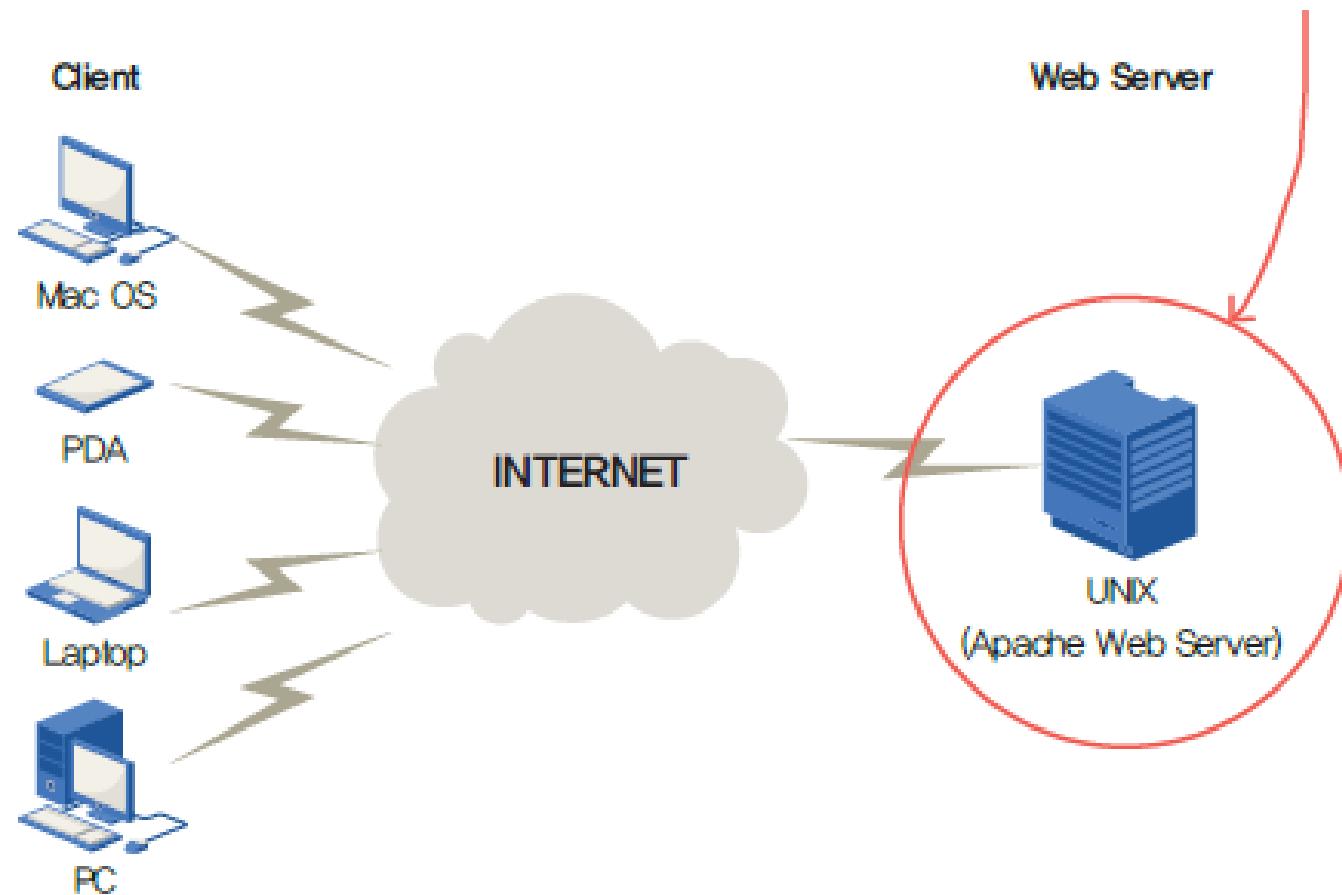
JSP





서버와 클라이언트

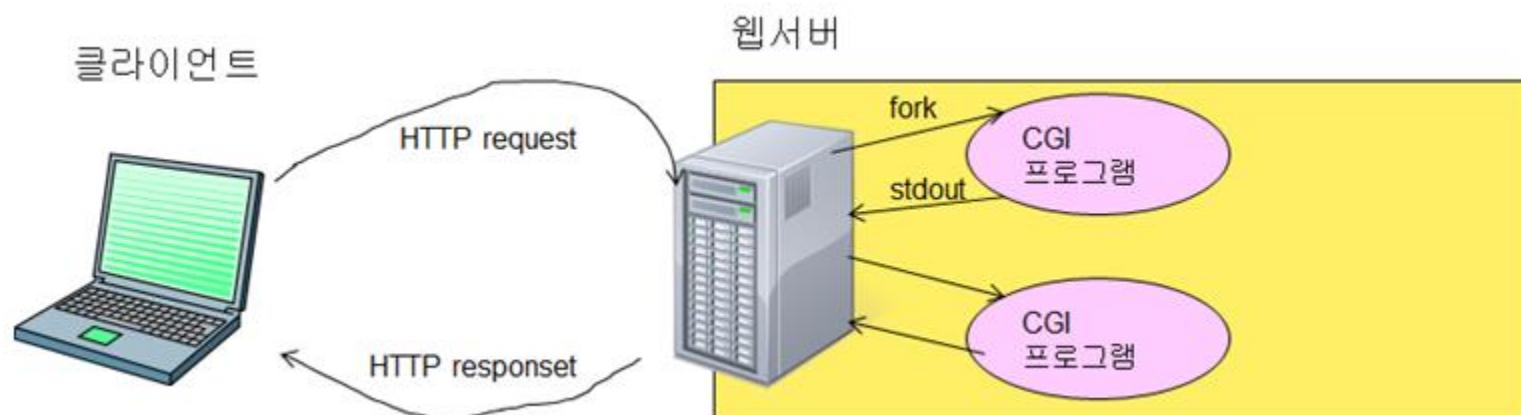
- 클라이언트는 서버에게 웹페이지를 요구
- 웹 서버는 웹페이지를 찾아서 클라이언트에게 전달





동적인 웹페이지

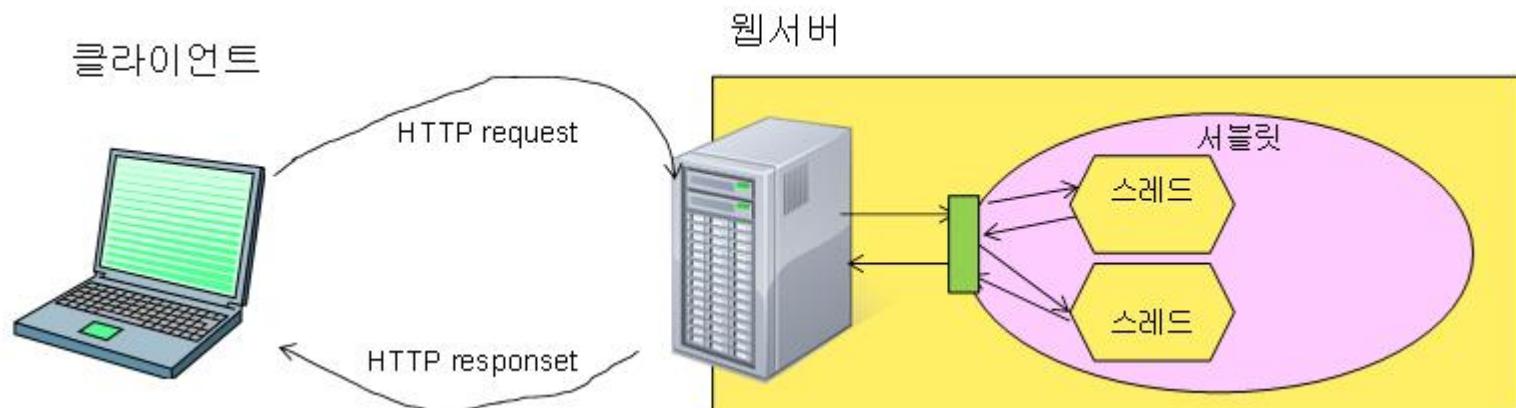
- 사용자의 요청에 따라서 웹페이지의 내용이 달라지는 페이지
- 게시판, 블로그, 방명록
- 서버 컴퓨터 안에 도우미 프로그램을 작성해 놓고 필요할 때마다 실행시켜서 결과를 얻은 후에 이것을 클라이언트 컴퓨터로 돌려주면 된다.





서블릿

- 서블릿: 자바를 서버쪽에서 사용하는 기술
- 서블릿의 장점
 - 효율적이다: 서블릿(servlet)은 스레드(thread)를 생성하여 각 요청을 서비스하므로 보다 효율적이다.
 - 편리하다: 자바 언어를 알고 있는 개발자
 - 강력하다: 자바 서블릿을 사용하면 기존의 CGI 프로그램으로는 아주 어려웠던 여러 가지 작업을 쉽게 처리할 수 있다.
 - 이식성이 있다: Apache에서 실행되는 서블릿은 MS IIS에서도 변경 없이 실행 가능





애플릿(Applet) & 서블릿(Servlet)

- <http://kyoe.egloos.com/768137>
- 웹브라우저를 통해 실행되는 자바프로그램은 2가지로 나눌 수 있다.
 - 클라이언트 측면에서 실행되는 '애플릿'
 - 서버 측면에서 실행되는 '서블릿'
- * 애플릿 *
 - 자바 언어로 구성된 작은 응용프로그램
 - 웹브라우저 상에서만 실행 될 수 있다.
 - html 문서에서 <applet code =" xxx.class"></applet> 태그를 이용해 애플릿을 상속받은 클래스'xxx'를 호출 받아 웹브라우저 상에 동작시킬 수 있다.
- * 서블릿 *
 - 자바로 구현된 CGI
 - 사용자의 정보를 입력 받아 적절한 HTML문서로 돌려주는 역할
 - 서블릿은 서블릿 규격에 맞게 구현해야 한다.



JSP

- 서블릿은 자바 언어 안에 HTML 태그를 넣어야 해서 복잡함
- JSP(JavaServer Pages)는 이러한 문제점을 해결하기 위하여 개발
- 자바를 기반으로 동적인 웹 페이지를 구축할 수 있는 서버-사이드 스크립트(server-side script) 언어
- 자바 서블릿을 기반으로 한다.

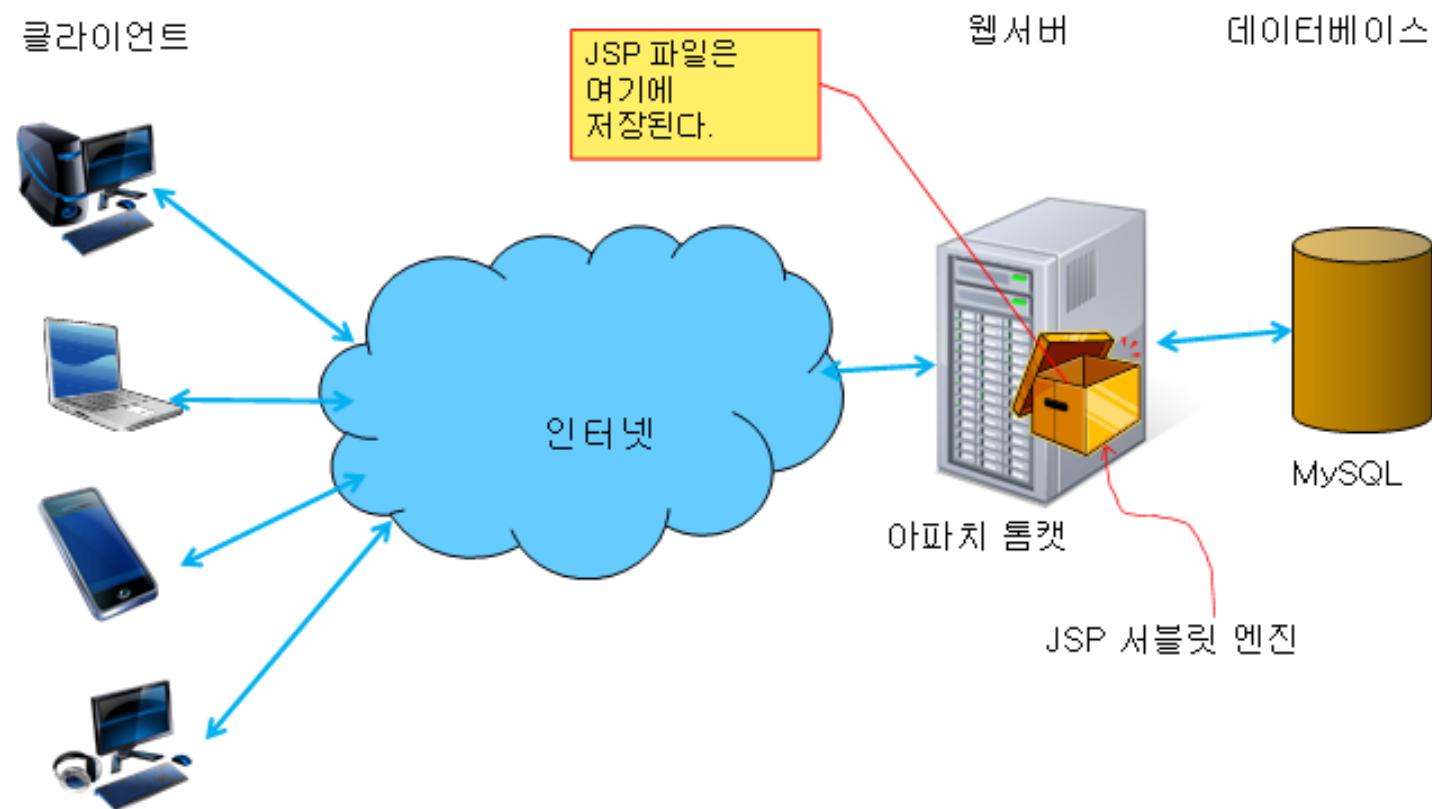
```
<html>
<head><title>Hello World</title></head>
<body>
Hello World!<br/>
<%
out.println("Your IP address is " + request.getRemoteAddr());
%>
</body>
</html>
```

필요할 때만 자바 코드를
HTML 문서에 삽입합니다.



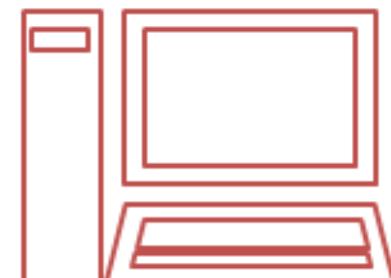
톰캣

- JSP 엔진의 일종
- 톰캣 = 웹 서버 + JSP 컨테이너



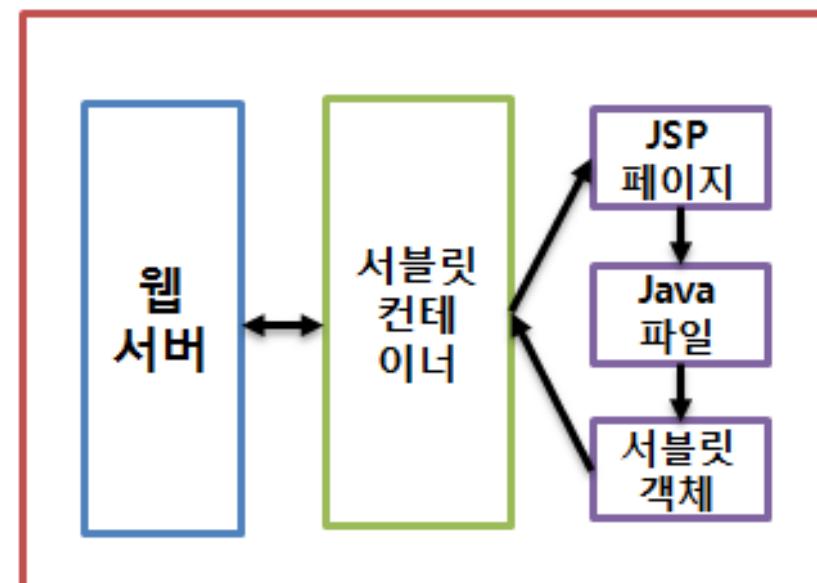


JSP 처리 방식



클라이언트

사용자 입력 데이터
(동적 웹 페이지 요청)
→
←
실행 결과
(HTML 페이지 응답)



서버



http://localhost:8080/ Apache Tomcat/8.0.23

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/8.0.23

The Apache Software Foundation
http://www.apache.org/



If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

Server Status

Manager App

Host Manager

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

Managing Tomcat
For security, access to the [manager](#) webapp is restricted. Users are defined

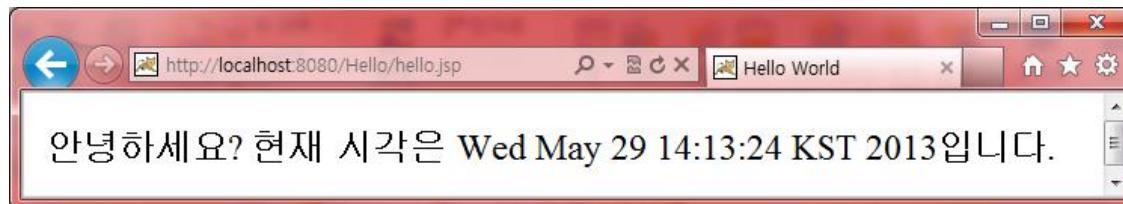
Documentation
[Tomcat 8.0 Documentation](#)

Getting Help
[FAQ](#) and [Mailing Lists](#)



실행 절차

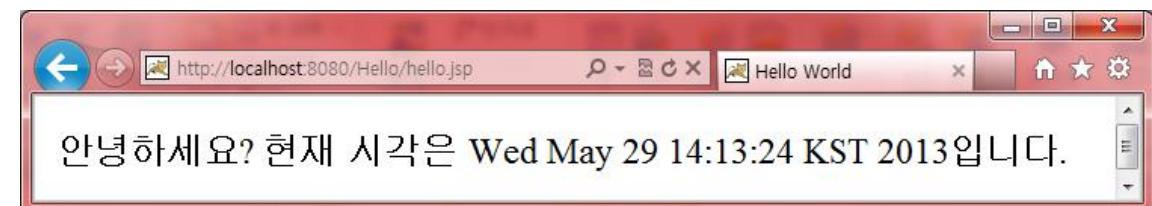
- Step 1: 아파치 톰캣을 시작한다.
- Step 2: 애플리케이션 폴더를 생성한다.
- Step 3: WEB-INF 작성(또는 복사)
- Step 4: hello.jsp 파일 작성(UTF-8로 저장)
- Step 5: 웹브라우저로 실행시키기(<http://localhost:8080>Hello/hello.jsp>)





hello.jsp

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
안녕하세요? 현재 시각은 <%= new java.util.Date() %>입니다.
</body>
</html>
```





hello_jsp.java

```
out.write("<html>\r\n");
out.write("<head>\r\n");
out.write("<title>Hello World</title>\r\n");
out.write("</head>\r\n");
out.write("<body>\r\n");
out.write("안녕하세요? 현재 시각은");
out.write( new java.util.Date() );
out.write("%>입니다.\r\n");
out.write("</body>\r\n");
out.write("</html>");
```

톰캣과 같은 JSP엔진은 JSP를 서블릿(Java안에 HTML 코드가 있다는 것이 특징)으로 변환하고 서블릿을 컴파일한 후 실행한다: 예제의 경우 “hello.jsp”를 “hello_jsp.java”로 변환한 결과를 보여준다.



JSP 수식

<%가 JSP를 나타내고 =은 자바 수식을 의미한다.

자바 언어의 어떤 수식이라도 넣을 수 있다.

JSP 가 종료되었다는 것을 의미하는 태그이다.

<%= *expression* %>



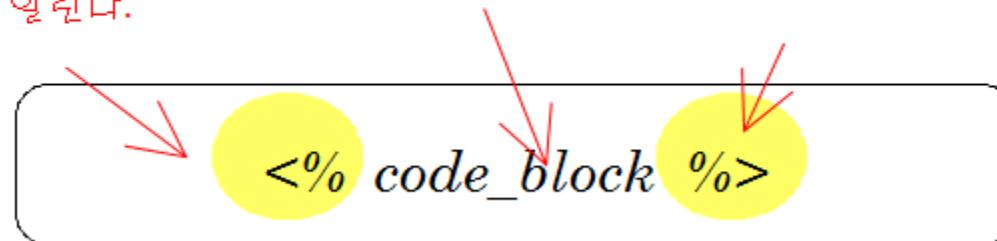
스크립틀릿(Scriptlet)

Scriptlet: A small script, especially a section of Java code embedded in Java Server Pages code.

<%가 자바 코드의 시작을 알린다.

어떠한 코드 블록도 가능하다.

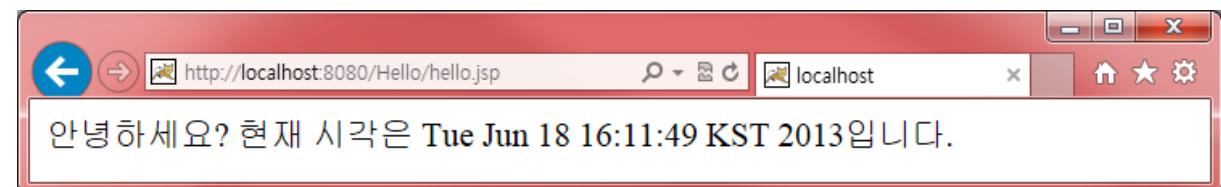
%>가 자바 코드의 종료를 알린다.





스크립틀릿 #1

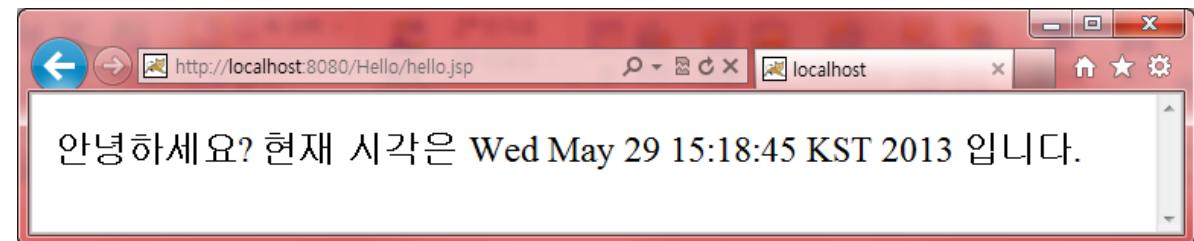
```
<!DOCTYPE html>
<HTML>
<BODY>
<%
    java.util.Date date = new java.util.Date();
%>
안녕하세요? 현재 시각은 <%= date %>입니다.
</BODY>
</HTML>
```





스크립틀릿 #2

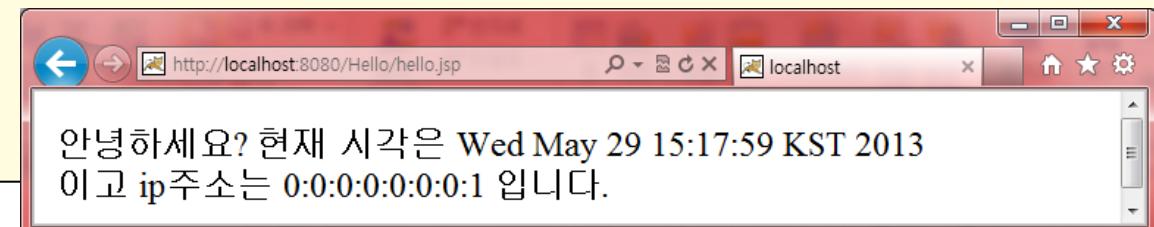
```
<!DOCTYPE html>
<HTML>
<BODY>
<%
    System.out.println( "날짜를 출력하여 본다." );
    java.util.Date date = new java.util.Date();
%>
안녕하세요? 현재 시각은
<%
    out.println( String.valueOf( date ) );
%>
입니다.
</BODY>
</HTML>
```





스크립틀릿 #3

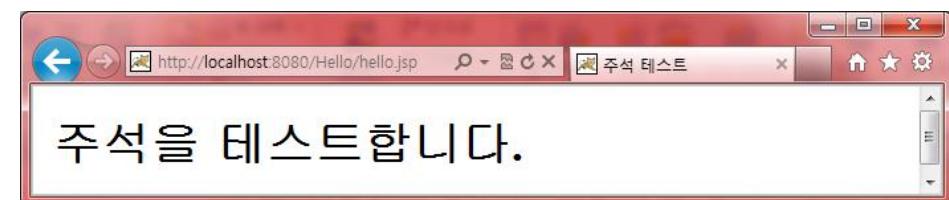
```
<!DOCTYPE html>
<HTML>
<BODY>
<%
    java.util.Date date = new java.util.Date();
%>
안녕하세요? 현재 시각은
<%
    out.println( date );
    out.println( "<BR>이고 ip주소는 " );
    out.println( request.getRemoteAddr() );
%>
입니다.
</BODY>
</HTML>
```





JSP 주석

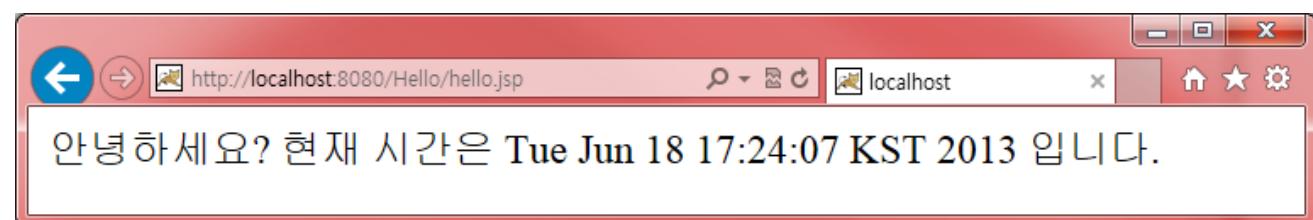
```
<!DOCTYPE html>
<html>
<head><title>주석 테스트</title></head>
<body>
<h2>주석을 테스트합니다.</h2>
<%-- 이 주석은 보이지 않습니다. --%>
</body>
</html>
```





JSP 지시어

```
<!DOCTYPE html>
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%
    Date date = new Date();
%>
안녕하세요? 현재 시간은 <%= date %> 입니다.
</BODY>
</HTML>
```



```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ include file="hello.jsp" %>
<%@ taglib uri="userLib" prefix="userTag" %>
```



JSP 선언

<%!>가 선언의 시작
을 알린다.

변수 선언이나 메소
드의 정의가 들어간
다.

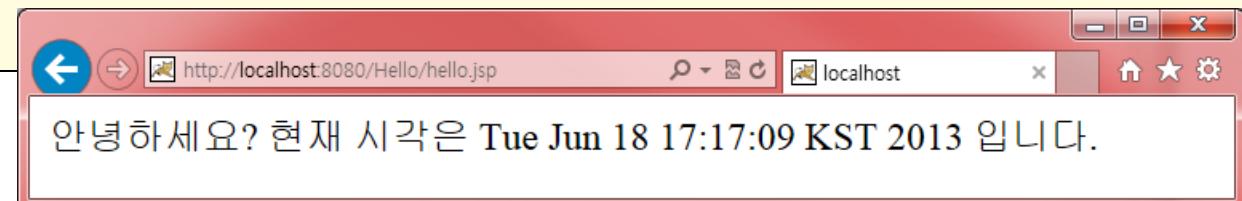
<%>가 선언의 끝을
알린다.

<%! Date date;... %>



JSP 선언

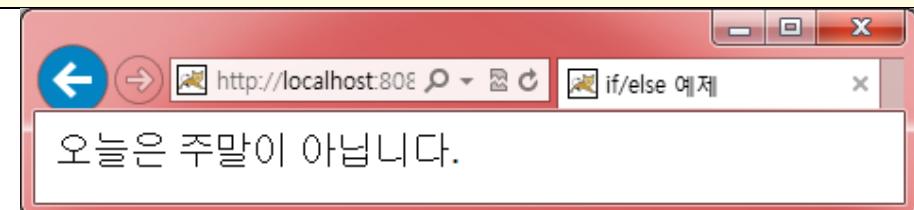
```
<!DOCTYPE html>
<HTML>
<BODY>
<%@ page import="java.util.*" %>
<%!
    Date date = new Date();
    Date getDate()
    {
        return date;
    }
%>
안녕하세요? 현재 시각은 <%= getDate() %> 입니다.
</BODY>
</HTML>
```





JSP 조건문

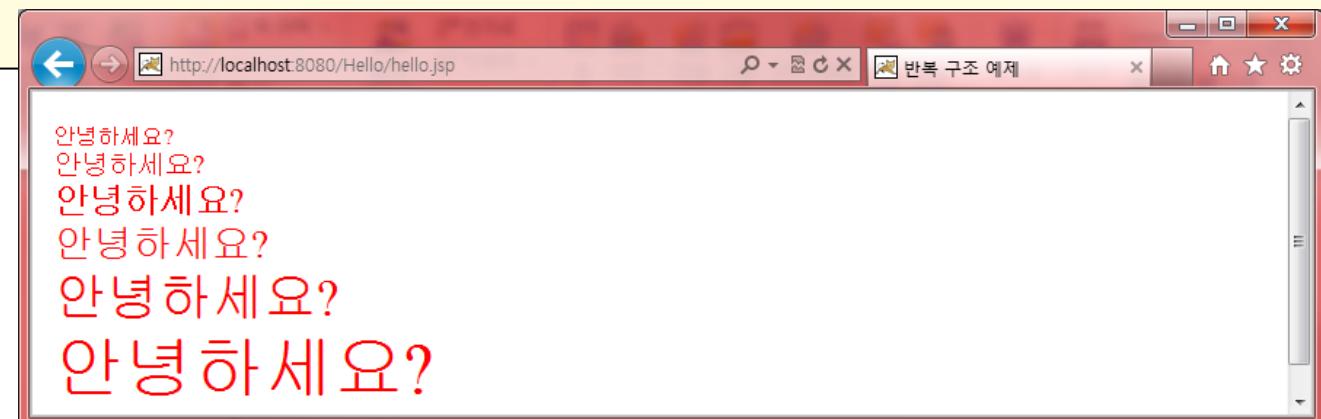
```
<%! int day = 3; %>
<html>
<head><title>if/else 예제</title></head>
<body>
<% if (day == 1 | day == 7) { %>
    <p> 오늘은 주말입니다.</p>
<% } else { %>
    <p> 오늘은 주말이 아닙니다.</p>
<% } %>
</body>
</html>
```





JSP 반복문

```
<%! int fontSize; %>
<html>
<head><title>반복 구조 예제</title></head>
<body>
<%for ( fontSize = 1; fontSize <= 6; fontSize++)%>
<font color="red" size="<%fontSize%>">
    안녕하세요?
</font><br />
<%}%>
</body>
</html>
```

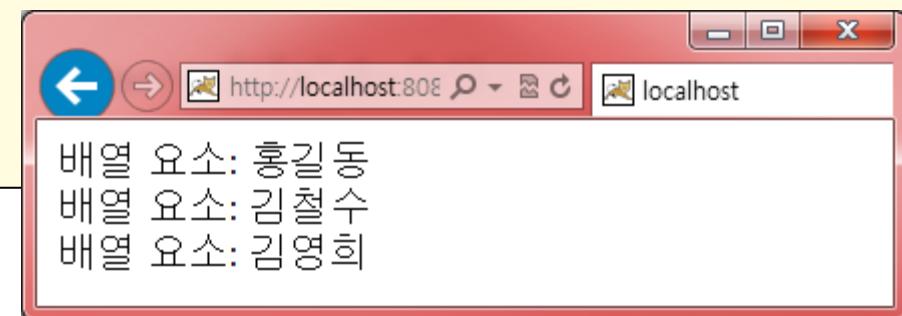




JSP 예제

```
<%@ page contentType="text/html; charset=utf-8" language="java" %>
<%
String[] array={"홍길동","김철수","김영희"};
%>

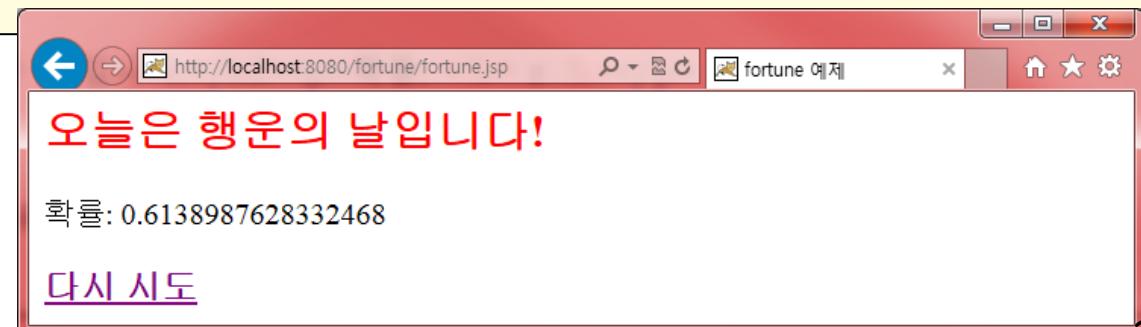
<html>
<body>
<%
int i=0;
for(i=0; i<array.length; i++)
{
    out.print("배열 요소: "+array[i]+"<br/>");
}
%>
</body>
</html>
```





JSP 난수 예제

```
<html>
<head><title>fortune 예제</title></head>
<body>
<%
    double r = Math.random();
    if (r > 0.60) {
%
        <h2 style="color:red">오늘은 행운의 날입니다!</h2><p>확률: <%= r
%></p>
<%
    } else {
%
        <h2>오늘은 평범한 날입니다.</h2><p>확률: <%= r %></p>
<%
    }
%
    <a href="<%= request.getRequestURI() %>"><h3>다시 시도</h3></a>
</body>
</html>
```





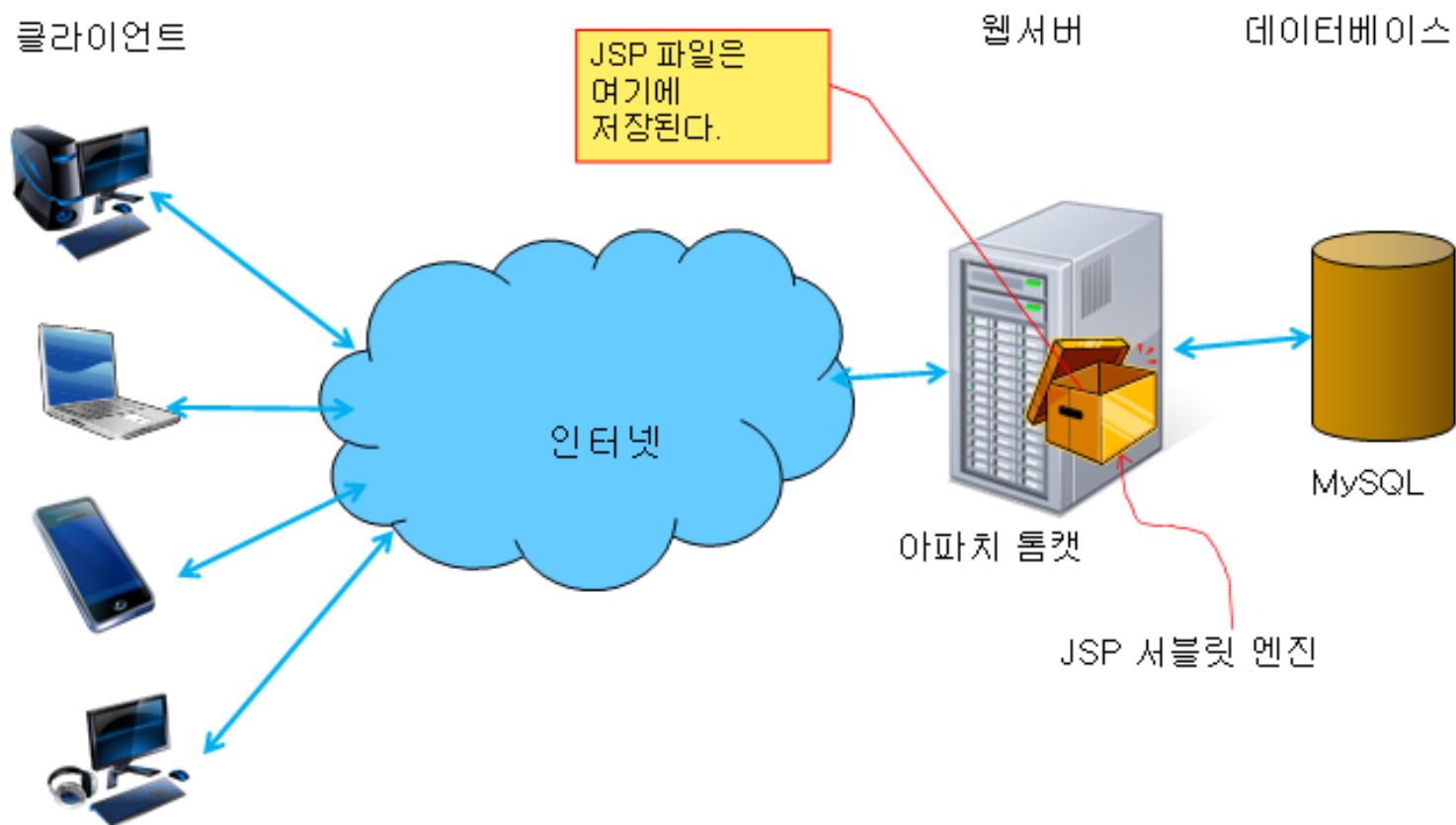
참고: JSP 내장 객체

내장 객체	설명
request	클라이언트의 요청 처리
response	클라이언트의 요청에 대한 응답
out	출력 스트림 처리
session	클라이언트의 세션 정보 처리
application	애플리케이션 정보 처리
pageContext	JSP 페이지의 내용(context)
config	JSP 페이지의 초기화(initialization) 매개변수 저장
page	현재의 JSP 페이지
exception	예외 처리



MySQL

- 공개 데이터베이스 서버





데이터베이스 예제: 온라인 서점

- MySQL 데이터베이스 드라이버 복사
 - mysql-connector-java-5.1.xx-bin.jar을 (톰캣 설치 디렉토리)/lib로 복사한다.
- 데이터베이스 생성
 - utf-8 버전의 명령어 행 클라이언트를 사용
- 웹 애플리케이션 생성
 - 톰캣 설치 디렉토리)/webapps에 ebookshop이라는 디렉토리를 생성



데이터베이스 쿼리 화면

order.jsp

```
<%@ page import="java.sql.*" %>
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<% request.setCharacterEncoding("utf-8"); %>

<html>
<head>
    <title>온라인 서점 예제</title>
</head>
<body>
    <h1>인터넷 프로그래머 문고</h1>
    <h3>제목을 입력하세요:</h3>
    <form method="post">
        책 제목: <input type="text" name="title"><br>
        <input type="submit" value="검색">
    </form>

    <%
        String title = request.getParameter("title");
    %>
```



데이터베이스 쿼리 화면

```
if (title != null) { %>
<%
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/book_db", "root", "1234");
    Statement stmt = conn.createStatement();
    String sqlStr = "SELECT * FROM book_table WHERE title LIKE ";
    sqlStr += "%" + title + "%";
    sqlStr += "ORDER BY title ASC";
    ResultSet rset = stmt.executeQuery(sqlStr);
%>    <hr>
<form method="post" action="orderproc.jsp">
    <table border=2>
        <tr>
            <th>주문</th>
            <th>저자</th>
            <th>제목</th>
            <th>가격</th>
            <th>수량</th>
        </tr>
```



데이터베이스 쿼리 화면

```
<%
    while (rset.next()) {
        int id = rset.getInt("id");
%>
    <tr>
        <td><input type="checkbox" name="id" value="<% id %>"></td>
        <td><%= rset.getString("author") %></td>
        <td><%= rset.getString("title") %></td>
        <td><%= rset.getInt("price") %>원</td>
        <td><%= rset.getInt("qty") %>권</td>
    </tr>
<%
    }
%>
</table>
```



데이터베이스 쿼리 화면

```
<br>
    <input type="submit" value="주문">
    <input type="reset" value="초기화">
</form>
    <a href="<%= request.getRequestURI() %>"><h3>다시
주문하기</h3></a>
<%
    rset.close();
    stmt.close();
    conn.close();
}
%
</body>
</html>
```



주문 처리 화면

order_proc.jsp

```
<html>
<head>
    <title>주문처리화면</title>
</head>
<body>
    <h2>주문해주셔서 감사합니다.</h2>
    <%
        String[] ids = request.getParameterValues("id");
        if (ids != null) {
    %>
    <%@ page import = "java.sql.*" %>
    <%
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn = DriverManager.getConnection(
            "jdbc:mysql://localhost:3306/book_db", "root", "1234");
        Statement stmt = conn.createStatement();
        String sqlStr;
        int recordUpdated;
        ResultSet rset;
    %>
```



주문 처리 화면

```
<table border=2>
  <tr>
    <th>저자</th>
    <th>제목</th>
    <th>가격</th>
    <th>수량</th>
  </tr>
<%
  for (int i = 0; i < ids.length; ++i) {
    sqlStr = "UPDATE book_table SET qty = qty - 1 WHERE id = " +
ids[i];
    recordUpdated = stmt.executeUpdate(sqlStr);
    sqlStr = "SELECT * FROM book_table WHERE id =" + ids[i];
    rset = stmt.executeQuery(sqlStr);
    while (rset.next()) {
      %>
      <tr>
        <td><%= rset.getString("author") %></td>
        <td><%= rset.getString("title") %></td>
        <td><%= rset.getInt("price") %>원</td>
        <td><%= rset.getInt("qty") %></td>
      </tr>
    <%
  }
}
```



주문 처리 화면

```
rset.close();
}
stmt.close();
conn.close();
}
%>
</table>
<a href="order.jsp"><h3>주문화면으로 돌아가기</h3></a>
</body>
</html>
```



실행 화면

인터넷 프로그래머 문고

제목을 입력하세요:

책 제목:

주문	저자	제목	가격	수량
<input checked="" type="checkbox"/>	수퍼맨	맨 오브 스틸	29000원	63권
<input checked="" type="checkbox"/>	스파이더맨	스파이더맨2	31000원	12권
<input type="checkbox"/>	아이언맨	아이언맨3	30000원	48권

주문해주셔서 감사합니다.

저자	제목	가격	수량
수퍼맨	맨 오브 스틸	29000원	62
스파이더맨	스파이더맨2	31000원	11

[주문화면으로 돌아가기](#)



Q & A

