CODING 101

WEB PROGRAMMING

01/24 16:00 PM

JSP Tag

JSP Tag

- **JSP**
  - HTML 이용하여 문서 작성
  - JSP Tag 이용하여 Java 코드 삽입
- **Servlet**
  - Java 언어를 이용하여 문서 작성
  - 출력 객체 (PrintWriter) 이용하여 HTML 코드 삽입

JSP Tag

➢ Scriptlet <% %>

- JSP 페이지에서 Java 언어 사용 위해 가장 많이 사용하는 태그
- 내부에 거의 모든 Java 코드 사용 가능

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <%
        int i = 0;
        while(true) {
            i++;
            out.println("2 x " + i + " = " + (2 * i) + "<br>");
    %>

    ========================= <br>

    <%
            if (i >= 9){
                break;
            }
        }
    %>

</body>
</html>
```

JSP Tag

CODING 101

- Declaration <%! %>
  - JSP 페이지에서 사용되는 변수, 메소드 선언할 때 사용
  - 전역 변수, 전역 메소드 선언

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

    <%!
        int i = 10;
        String str = "Hello World!";

        public int add(int a, int b){
            return a + b;
        }
    %>

    <%
        //jsp에서는 PrintWriter 객체 out은 서버에서 자동으로 생성해 줌 - (PrintWriter-브라우저에 보여 줌) : 내장 객체
        //PrintWriter out = response.getWriter(); //내장 객체 response와 out
        out.println("i의 값은: " + i + "<br>");
        out.println("str의 값은: " + str + "<br>");
        out.println("30과 50의 합은: " + add(30, 50));
    %>

</body>
</html>
```

JSP Tag

➢ Expression <%= %>

- JSP 페이지에서 사용되는 변수의 값, 메소드 호출 결과 값을 출력하기 위해 사용
- 결과값의 데이터 타입은 String
- 세미콜론 사용 X

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

    <%!
        //전역 변수로 선언 - class expression_jsp 내에서 선언
        String str = "Hello";
        double d = 3.1415;
    %>

    <%
        //지역 변수로 선언 -  _jspService() 내에서 선언
        int age = 30;
        String name = "홍길동";
    %>

    <!-- 표현식에는 세미콜론 사용하지 않음 -->
    이름: <%= name %> <br>
    나이: <%= age %> <br>

</body>
</html>
```

JSP Tag

➢ Comments <%-- --%>

- 프로그램 설명 등의 목적으로 사용
- HTML : <!-- --> (브라우저에서 페이지 소스 보기 하면 주석도 표기 O)
- JSP : <%-- --%> (브라우저에서 페이지 소스 보기 하면 주석은 표기 X)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

    <%
        int a = 100;
        //스크립트 안에서는 자바 주석 사용!
        /* 자바 주석입니다 */
    %>

    <!-- html 주석 부분 -->
    <h2>안녕하세요</h2>

    <%-- jsp 주석 부분, 여기는 페이지 소스코드에도 보이지 않음 --%>
    <h2>반갑습니다</h2>

</body>
</html>
```

JSP Tag

➢ Directive <%@ %>

- JSP 페이지의 전체적인 속성을 지정할 때 사용

① page

- JSP 페이지에 대한 정보 지정
- JSP가 생성하는 문서 타입, 출력 버퍼 크기, 에러 페이지 등 페이지에서 필요로 하는 정보 설정
- 페이지 directive에 선언될 수 있는 속성들
  - ✓ language : 언어 지정 (java만 지정 가능)
  - ✓ import : 패키지 import
  - ✓ errorPage : 에러 발생 시 미리 만들어 둔 에러 페이지 (view file) 호출
  - ✓ contentType : JSP가 생성하는 문서 타입 (text/html이라고 선언)
  - ✓ pageEncoding : 출력 문자 인코딩 (utf-8로 지정)
  - ✓ trimDirectiveWhitespaces : Directive나 Scriptlet 코드로 인해 생긴 공백 제거

② include

- JSP 페이지의 특정 영역에 다른 문서 포함

```jsp
<%@page import="java.util.Calendar"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page trimDirectiveWhitespaces = "true" %> <%-- 페이지 소스 보기에서 자바 코드 지워진 흰 공백 안 보여 줌 --%>
<%
    Calendar cal = Calendar.getInstance();%> <%-- abstract class이기 때문에 --%>


<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

    오늘은
    <%= cal.get(Calendar.YEAR) %> 년 
    <%= cal.get(Calendar.MONTH) + 1 %>월 
    <%= cal.get(Calendar.DATE) %>일입니다. <br>

    현재 시각: <%= new java.util.Date() %>

</body>
</html>
```

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page trimDirectiveWhitespaces="true" %>
<%@ include file = "include/header.jsp" %>


<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <h2>방문자 수: <%= visit %></h2> <!-- header.jsp 인클루드해서 변수 사용 가능 -->
    <h3>바디 부분</h3> <br>

    <%@ include file = "include/footer.jsp" %>
</body>
</html>
```

실습

JSP Object

JSP Object

➢ JSP 내장 객체

- 개발자가 JSP 파일 내에 객체를 생성하지 않고 바로 사용할 수 있는 객체
- JSP 내장 객체는 Servlet으로 변환될 때 자동으로 객체가 생성됨

➢ JSP 내장 객체 종류

• request	javax.servlet.http.HttpServletRequest
• response	javax.servlet.http.HttpServletResponse
• out	javax.servlet.jsp.JspWriter
• session	javax.servlet.http.HttpSession
• application	javax.servlet.ServletContent
• pageContext	javax.servlet.jsp.PageContext
• page	javax.servlet.jsp.HttpJspPage
• config	javax.servlet.ServletConfig
• exception	java.lang.Throwable

Request 객체

➢ Request 객체

- Request : 웹 브라우저를 통해 서버에 어떤 정보를 요청하는 것
- Request 객체 : 이러한 요청 정보가 담기고 관리되는 곳

➢ Request 객체 제공 기능

- 클라이언트(웹 브라우저)와 관련된 정보 읽기
- 서버와 관련된 정보 읽기
- 클라이언트가 전송한 요청 파라미터 읽기
- 클라이언트가 전송한 쿠키 읽기

➢ Request 객체 관련 주요 Method

➢ Request 객체 관련 주요 Method

Method Name	Return Type	Function
getContextPath()	String	웹 어플리케이션의 컨텍스트 루트의 경로 리턴
getMethod()	String	웹 브라우저가 정보를 전송할 때 사용한 요청 방식을 리턴 (get/post)
getServerName()	String	연결할 때 사용한 서버 이름 리턴
getServerPort()	int	서버가 실행중인 포트 번호를 리턴
getRequestURL()	String	요청 URL 리턴
getRequestURI()	String	요청 URI 리턴
getRemoteAddr()	String	웹 서버에 연결한 클라이언트의 IP주소를 리턴
getProtocol()	String	해당 프로토콜을 리턴
getParameter(String name)	String	이름이 name인 파라미터 값을 구함 (존재하지 않을 경우 null 리턴)
getParameterValues(String name)	String []	이름이 name인 모든 파라미터 값들을 배열로 구함 (존재하지 않을 경우 null 리턴)
getParameterNames()	java.util.Enumeration	서버에 전송한 파라미터의 이름 목록을 리턴
getParameterMap()	java.util.Map	웹 브라우저가 전송한 파라미터의 Map 리턴 cf) Map은 <파라미터 이름, 파라미터 값> 쌍으로 구성

Response 객체

➢ Response 객체

- Response : Request에 응답하는 것
- Response 객체 : 이러한 응답 정보를 가지고 있는 객체

➢ Response 객체 관련 주요 Method

Method Name	Function
getCharacterEncoding()	응답할 때의 문자 인코딩 형태 리턴
addCookie(Cookie c)	쿠키 지정
sendRedirect(URL)	지정한 URL로 리다이렉트

PrintWriter 객체 (out)

- ✓ JSP 페이지가 생성하는 모든 내용은 out 기본 객체를 통해 전송
- ✓ JSP 페이지 내에서 사용하는 비스크립트 요소들(HTML코드와 텍스트)도 out 객체에 전달
- ✓ 값을 출력하는 표현식(expression)의 결과값도 out 객체에 전달

실습