

## ■ 주석이란 ?

- 주석은 프로그램 소스에 텍스트로 된 간단한 설명문을 넣는 것을 말한다.
- C 언어를 비롯한 대부분의 프로그램 언어가 주석을 사용하기 위한 문법을 제공하고 있다.
- JSP는 특성상 자바, HTML, JSP 코드가 섞여 있으므로 주석도 혼용해서 사용한다.
- **HTML 주석 : 클라이언트로 전달되는 주석**
  - 일반적인 HTML 문서에서 사용 가능한 주석으로 화면에는 보이지 않지만 브라우저 소스보기를 하면 내용이 노출됨.

```
<!-- 주석입니다. -->
```

- **JSP 주석 : 클라이언트로 전달되지 않는 주석**
  - JSP 파일에서만 사용 가능한 주석으로 브라우저 소스보기를 해도 내용이 노출되지 않음.

```
<%-- 주석 --%>
```

### ■ 지시어(Directives)란?

- 지시어(Directives)는 JSP 파일의 속성을 기술하는 JSP 문법.
- JSP 컨테이너에게 해당 페이지를 어떻게 처리해야 하는지 전달하기 위한 내용을 담고 있다.
- 지시어는 크게 page, include, taglib으로 나눌 수 있으며, 각각에서 다루는 속성이 다르다.

### 1. page 지시어

- page 지시어는 현재 JSP 페이지를 컨테이너에서 처리하는 데 필요한 각종 속성을 기술하는 부분.
- 보통 JSP 페이지 맨 앞에 위치함.

```
<%@ page 속성1="속성값1" 속성2="속성값2" ... %>
```

- 여러 줄에 나누어 작성할 수도 있음.

```
<%@ page contentType="text/html;charset=UTF-8"  
    import="javax.sql.*, java.util.*" errorPage="error.jsp"%>  
<%@ page import="java.util.*" %>
```

## 02. 지시어

- 이클립스 개발 도구를 이용해 jsp 파일을 생성하는 경우 기본적인 page 지시어는 자동 생성됨.
- 필요에 따라 속성을 추가해 사용

[표 5-1] page 지시어 속성

속성	설명	기본 설정 값
language	스크립트 언어의 유형을 정한다.	java
import	JSP 내에서 사용할 외부 자바 패키지나 클래스의 불러오기(import)를 정한다.	-
session	세션의 사용 유무를 정한다.	true
buffer	버퍼의 크기를 정한다.	8KB
autoFlush	버퍼의 내용을 자동으로 비운다.	true
isThreadSafe	단일 스레드 모델을 사용함으로써 동시성 제어 여부를 정한다.	true
info	JSP 페이지에 대한 설명이다.	-
errorPage	현재 페이지에서 오류가 발생할 경우 호출될 페이지를 지정한다.	-
isErrorPage	오류만을 처리하는 페이지로 지정한다.	false
contentType	MIME 형식 지정 및 캐릭터셋을 설정한다.	text/html;charset=ISO-8859-1
pageEncoding	contentType과 동일한 기능을 한다.	ISO-8859-1
extends	현재 JSP 페이지를 특정 클래스를 상속한 클래스로 작성한다.	[표준스펙] javax.servlet.jsp.HttpJspPage [톰캣구현] org.apache.jasper.runtime.HttpJspBase

## 02. 지시어

### ■ Page 지시어와 JSP의 한글 처리

- page 지시어에서 중요한 부분 중 하나는 한글 처리 부분임.
- JSP 에서는 다음과 같이 3단계로 캐릭터셋을 설정함.

[표 5-2] JSP 에서의 캐릭터셋 관련 설정 항목

우선순위	설정 방법	비고
1	server.xml 파일에서 정의한다.	Connector 설정에 URI 요청에 대한 캐릭터셋을 지정할 수 있다. 작성한 소스를 다른 서버에서 실행할 경우, 해당 서버 설정에 따라 한글이 깨질 수도 있다.
2	각 JSP 파일, page 지시어의 pageEncoding 속성에서 정의한다.	페이지 단위의 설정이므로 애플리케이션 설정과 무관하게 한글을 처리할 수 있다. 단, 서버가 JSP 2.0 스펙을 지원하지 않는 예전 버전이라면 pageEncoding 속성을 인식하지 못해 한글이 깨질 수 있다.
3	각 JSP 파일, page 지시어의 contentType 속성에서 정의한다.	클라이언트에 대한 응답 결과를 지정하는 부분으로 브라우저에게 전달할 캐릭터셋을 지정할 수 있다.

- 위 설정에서 캐릭터셋 설정을 찾지 못할 경우 ISO8859-1을 적용한다.
- 페이지 지시어에 다음과 같이 한글 속성 설정
  - pageEncoding="UTF-8", contentType="text/html; charset=UTF-8"

## 02. 지시어

### ■ import

- import 는 JSP 스크립트 부분에서 자바 클래스를 사용하는 경우 해당 클래스의 패키지에 대한 import 설정으로 기본적으로 자바에서와 동일하다.
- 다만 패키지 import 구분을"," 을 이용하거나 라인 단위로 작성해야 한다.
- 다음은 page 지시어에서 import 속성의 다양한 사용 예이다.

```
<%@ page import="java.sql.*,java.util.*" %>
```

```
<%@ page import="java.sql.*" %>
```

```
<%@ page import="java.util.*" %>
```

```
<%@ page  
    import="java.sql.*,  
    java.util.*"  
%>
```

## 1. JSP 액션의 종류

- JSP 액션은 JSP 고유 기능으로 빈즈 클래스 연동 및 동적 페이지 관리를 위한 기능을 제공함.
- `<jsp:action_name attribute="value" />` 형태를 가짐.
- 주로 사용하는 액션은 useBean, get/setProperty 이며 자바 클래스와의 연동을 위해 사용함.

→ 7장. JSP와 자바빈즈

- 액션(Action)은 JSP 주요 구성요소 중 하나로 다음과 같은 기능을 지원한다.
  - JSP 페이지간 흐름 제어
  - 자바 애플릿 지원
  - 자바 빈즈 컴포넌트와 JSP 상호작용 지원
- 특히 useBean 액션은 JSP에서 자바 빈즈 클래스와의 연동을 지원해주는 액션으로 잘 알아둘 필요가 있다.
- include 액션은 단순히 페이지를 포함하는 것 뿐만 아니라 파라미터를 포함될 페이지로 전달하는 것이 가능함.
  - 사용 예 ) `<jsp:param name="user" value="홍길동" />`

## 03. 액션

- 대표적인 액션은 표5-3과 같음.

[표 5-3] 대표적인 액션의 종류

액션	사용 예	기능
include	<code>&lt;jsp:include page="xx.jsp" /&gt;</code>	다른 페이지를 현재 페이지에 포함시킨다.
forward	<code>&lt;jsp:forward page="xx.jsp" /&gt;</code>	현재 페이지의 제어를 다른 페이지로 전달한다.
useBean	<code>&lt;jsp:useBean scope="page" id="cls" class="xx.MyBean" /&gt;</code>	xx패키지의 MyBean 클래스를 cls라는 이름으로 page 범위에서 사용할 것을 선언한다.
setProperty	<code>&lt;jsp:setProperty name="cls" property="xxx" /&gt;</code>	useBean으로 선언된 빈즈 클래스의 setxxx() 메서드를 호출한다.

액션	사용 예	기능
getProperty	<code>&lt;jsp:getProperty name="cls" property="xxx" /&gt;</code>	useBean으로 선언된 빈즈 클래스의 getxxx() 메서드를 호출한다.
plugin	<code>&lt;jsp:plugin type="applet/bean" code="class"&gt; &lt;/jsp:plugin&gt;</code>	애플릿이나 빈즈 클래스를 플러그인 형태로 로딩한다.
param	<code>&lt;jsp:param name="user" value="홍길동" /&gt;</code>	include, forward 액션에서 사용할 수 있는 파라미터를 설정한다.

### 5. useBean 액션

- 액션에서 가장 중요한 부분으로, JSP 빈즈를 다루는 7장에서 자세히 살펴볼 것이다.
- 여기서는 기본 표기 방법만을 살펴보기로 한다.
- 사용법

```
<jsp:useBean id="변수_이름" class="빈즈 클래스_이름"/>  
<jsp:setProperty name="변수_이름" property="속성_이름"/>  
<jsp:getProperty name="변수_이름" property="속성_이름"/>
```

- useBean 액션은 빈즈 클래스를 사용하기 위한 구문이며 class 에 지정된 자바 빈즈 클래스를 id 라는 이름으로 사용할 수 있도록 해준다.
- get/setProperty 액션은 브라우저에서 빈즈 클래스의 멤버 변수로 값을 저장하거나 가져오기 위한 구문이다.
- get/setProperty는 빈즈 클래스의 getter/setter 메서드와 연동된다.



## 2. 표현식

- 표현식(Expression)은 이미 여러 소스를 통해 많이 살펴본 것처럼 `<%= %>`를 사용해서 간단한 데이터 출력이나 메서드 호출 등에 이용한다.
- 코드 마지막에 `;(세미콜론)`을 사용하지 않는다는 것에 주의하도록 한다.
- 표현식은 결국 `out.println()` 으로 변환되기 때문에 `out.println()`의 인자로서 적합한 형태로 사용해야 함.
- 메서드 호출, 변수출력, 사칙연산 및 문자열 결합 등이 가능하다.
- 표현식 보다는 표현언어(Expression Language) 사용을 권장한다. → 10장
- 사용 예)

메서드 호출 : `<%= calculator() %>`

변수 출력 : `<%= result %>`

사칙 연산과 문자열 결합 : `<%= "i+2="+(i+2)+"입니다" %>`

## 1. 스크립트릿이란?

- 스크립트릿(Scriptlet) 은 JSP 문서 내에서 자바 코드를 기술할 수 있는 부분으로 JSP의 가장 큰 특징 중 하나.
- 초기 JSP 발전에 큰 기여를 하였으나 지금은 JSP내에서 스크립트릿 사용은 권장되지 않음.
- JSP 내에서 화면과 프로그램이 섞여 있으면 유지보수가 힘들고 디자이너들과의 협업이 어려움.
- MVC 패턴에 따라 웹 프로그램을 개발하게 되면 JSP는 View의 역할을 하게 되고 표현언어, JSTL, 커스텀 태그 라이브러리, JSP 빈즈가 주로 사용된다.
- 그러나 JSP의 가장 큰 특징이며 유용한 부분이기도 하므로 자유자재로 사용할 수 있어야 함.
- UI(AWT, Swing 등) 을 제외한 거의 모든 자바 클래스 라이브러리를 사용한 프로그래밍이 가능함.
- 사용 예)

```
<%  
    // 로컬 변수 선언이나 프로그램 로직이 올 수 있다.  
    String str = "test";  
    for (int i=0; i < 10; i++) {  
        out.println(i);  
    }  
%>
```