4장. JSP의 구성요소와 스크립팅요소

1. JSP 구성 요소 소개

- 스크립팅 요소 (Scripting Element)
  - 1) 지시문 (Directive)
    - JSP 엔진 및 컨테이너, 즉 Tomcat에게 현재의 JSP 페이지 처리와 관련된 정보를 전달하는 목적으로 활용 (6장)

```
<%@ page contentType="text/html;charset=utf-8"%>
```

  - 2) 스크립트릿 (Scriptlet)
    - 가장 많이 활용되는 JSP 구성 요소로서 JSP 페이지 내에서 코드 구현을 위해 사용

```
<%
    for (int i=0; i<=10; i++) {
        out.println("Hello World" + i + "<br/>");
    }
%>
```

1. JSP 구성 요소 소개

- 스크립팅 요소 (Scripting Element)
  - 3) 표현식 (Expression)
    - 간단한 데이터 출력이나 메소드 호출을 통한 데이터 출력을 위해 활용
    - 코드 마지막에 세미콜론(;)이 없다는 것에 주의

```
<%= result + resultSum() %>
```

  - 4) 선언 (Declaration)
    - JSP 페이지 전체에서 활용할 변수 및 메소드를 선언

```
<%!
    String name="Gildong Hong";
    public boolean isExist() {
       return true;
    }
%>
```

1. JSP 구성 요소 소개

- 스크립팅 요소 (Scripting Element)
  - 5) 주석 (Comment)
    - 코드 상에 부가적인 설명을 덧붙이기 위해 사용
    - JSP에서 활용할 수 있는 주석의 종류는 총 3가지

```
<%-- This is JSP Comment --%>
```

1. JSP 구성 요소 소개

- XML 태그
  - 1) 액션 태그 (Action Tag)
    - JSP 페이지간의 흐름 제어 및 자바 빈즈 컴포넌트와 JSP의 상호작용을 위해 사용

```
<jsp:include page="heading.jsp" />
```

  - 2) JSTL (Java Standard Tag Library)
    - 개발자가 JSP 페이지를 구성할 때 많이 활용하는 기능을 모아서 XML 태그들로 구성하여 라이브러리화 해 놓은 것

```
<c:if test="${1+1==2}">
    Always true.
</c:if>
```

  - 3) 커스텀 태그 (Custom Tag)
    - JSP 페이지 개발자가 많이 활용되는 로직 처리 및 프리젠테이션 기능을 직접 개발하여 활용하는 XML 태그

```
<user:getUserShoppingList userId="13" />
```

1. JSP 구성 요소 소개

- 독립 언어
  - 1) 표현 언어 (Expression Language, EL)
    - 서버 측의 다양한 상태정보를 개발자로 하여금 손쉽게 프리젠테이션할 수 있도록 해주는 용도로 개발된 언어
    - JSP의 기본 문법을 보완하는 역할로서 서버가 지니고 있는 다양한 값들에 대한 표현을 하기 위한 언어
    - 스크립팅 요소의 표현식(Expression)과 혼동하면 안된다.
    - 보통 JSTL 및 커스텀 태그와 함께 사용된다.
    - ${ .... } 로 표기된다.

```
${colorBean.red}

<c:if test="${1+1==2}">
    Always true.
</c:if>
```

2. 스크립팅 요소 (Scripting Element)

- 스크립트릿 (Scriptlet)
  - JSP의 파워가 강한 이유: 스크립트릿을 활용하여 JSP 페이지 내에 Java 코드를 넣을 수 있기 때문
  - <%과 %> 사이에 Java 코드 위치
  - 코드 블록내의 각 문장은 Java 문법을 따르기 때문에 반드시 마지막에 세미콜론(;)을 넣어야 한다.

```
<%
    Java 코드 1;
    Java 코드 2;
    Java 코드 3;
    …
    …
%>
```

2. 스크립팅 요소 (Scripting Element)

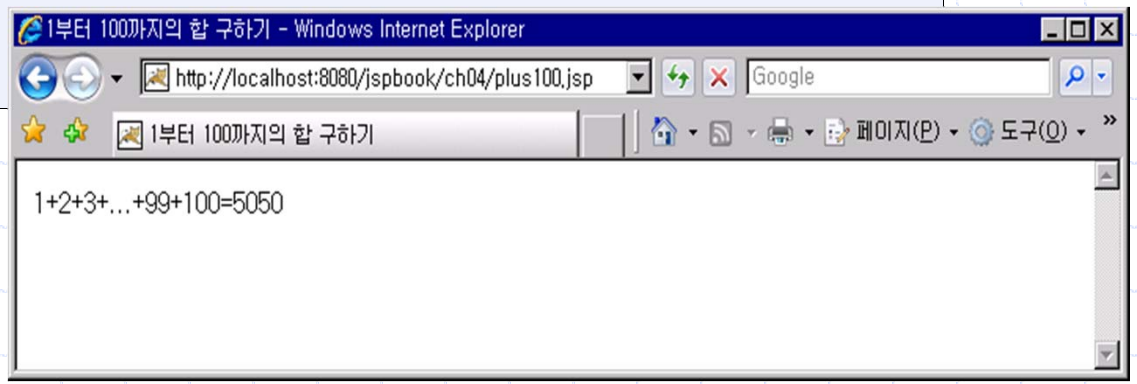
스크립트릿 (Scriptlet)

- 1부터 100까지의 합을 구하는 JSP 페이지를 스크립트릿 I

[예제 4.1] **jspbook\ch04\plus100.jsp**

```
01  <%@ page contentType="text/html;charset=utf-8" %>
02  <html>
03  <head>
04  <title>1부터 100까지의 합 구하기</title>
05  </head>
06  <body>
07  <%   //스크립트릿을 나타내는 기호
08      int sum = 0;
09      for (int i = 1; i <= 100; i++) {
10        sum = sum + i;
11      }
12  %>  // 스크립트릿을 나타내는 기호
13  1+2+3+...+99+100=<%= sum %>  //표현식을 이용한 sum 출력
14  </body>
15  </html>
```

Java 코드 (문장별로 세미콜론 기입요)



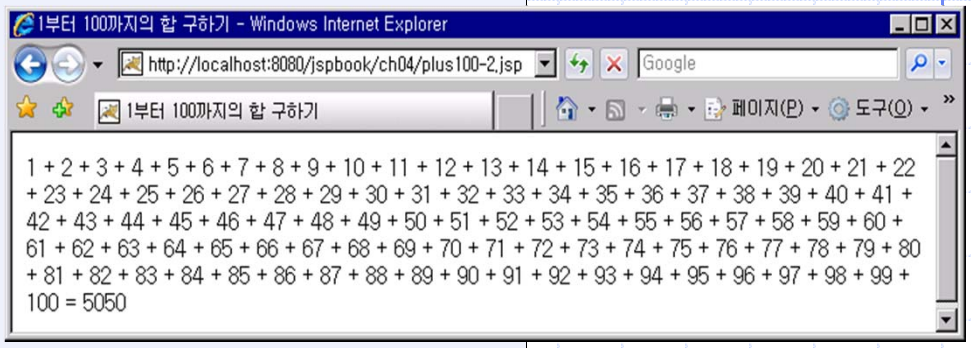
2. 스크립팅 요소 (Scripting Element)

스크립트릿 (Scriptlet)

- 1부터 100까지의 합을 구하는 JSP 페이지를 스크립트릿 II-1

[예제 4.2] jspbook\ch04\plus100-2.jsp

```
01 <%@ page contentType="text/html; charset=utf-8" %>
02 <html>
03 <head>
04 <title>1부터 100까지의 합 구하기</title>
05 </head>
06 <body>
07 <%                                          ┐
08 int sum = 0;                                │
09 for (int i = 1; i <= 100; i++) {            ├ 스크립트릿 블록 1
10    sum = sum + i;                           │
11 }                                           │
12 %>                                         ┘
13 <%                                          ┐
14 for (int i = 1; i <= 99; i++) {             ├ 스크립트릿 블록 2
15 %>                                         ┘
16 <%= i %> +      // 표현식을 이용한 출력 1
17 <%                                          ┐
18 }                                           ├ 스크립트릿 블록 3
19 %>                                         ┘
20 <%= 100 %> = <%= sum %>   // 표현식을 이용한 출력 2
21 </body>
22 </html>
```



2. 스크립팅 요소 (Scripting Element)

- 스크립트릿 (Scriptlet)
  - 1부터 100까지의 합을 구하는 JSP 페이지를 스크립트릿 II-2

[예제 4.3] jspbook\ch04\plus100-3.jsp

```
01 <%@ page contentType="text/html;charset=utf-8" %>
02 <html>
03 <head>
04 <title>1부터 100까지의 합 구하기</title>
05 </head>
06 <body>
07 <%
08     int sum = 0;
09     for (int i = 1; i <= 100; i++) {
10       sum = sum + i;
11     }
12 %>
13 <%
14     for (int i = 1; i <= 99; i++) {
15       out.print(i + " + ");   // out 객체를 활용한 출력
16     }
17     out.print(100 + " = " + sum); // out 객체를 활용한 출력
18 %>
19 </body>
20 </html>
```

스크립트릿 블록 1

스크립트릿 블록 2

2. 스크립팅 요소 (Scripting Element)

- 스크립트릿 (Scriptlet)
  - 복잡한 스크립트릿 기호 남용을 더욱 줄이기

```
<%
    int sum = 0;
    for (int i = 1; i <= 100; i++) {
        sum = sum + i;
    }
    for (int i = 1; i <= 99; i++) {
        out.print(i + " + ");
    }
    out.print(100 + " = " + sum);
%>
```

  - 이 책의 후반부에서 다루는 JSTL 및 EL (Expression Language)과 커스텀 태그를 활용하면 스크립트릿의 남발을 더욱 잘 방지할 수 있다.

2. 스크립팅 요소 (Scripting Element)

- 표현식 (Expression)
  - 간단한 데이터 출력을 위하여 사용
  - **<%=** 로 시작하여 **%>**로 끝나며 이 둘 사이에 출력해야 할 값을 적는다.
  - 출력할 수 있는 값
    - 단순한 문자열 및 숫자
    - 변수
    - 수식
    - 메소드 호출
  - 마지막에 세미콜론(;)을 붙이지 않는다

```
<%= 값 %>
```

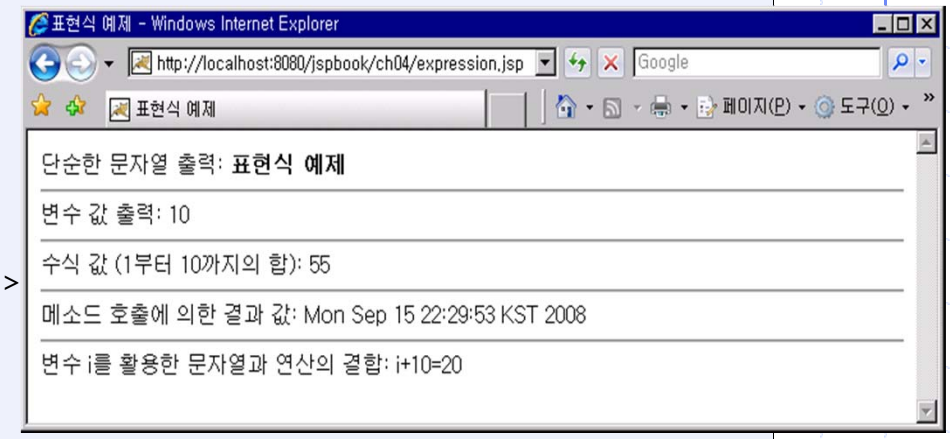
2. 스크립팅 요소 (Scripting Element)

표현식 (Expression)

- 관련 예제

[예제 4.4] jspbook\ch04\expression.jsp

```
01  <%@ page contentType="text/html;charset=utf-8" %>
02  <html>
03  <head>
04  <title>표현식 예제</title>
05  </head>
06  <body>
07    단순한 문자열 출력: <%= "<b>표현식 예제</b>" %>
08  <hr/>
09  <%
10    int i = 10;
11    java.util.Calendar cal = java.util.Calendar.getInstance();  // Calendar 객체 생성하여 cal 변수에 할당
12  %>
13  변수 값 출력: <%= i %>
14  <hr/>
15  수식 값 (1부터 10까지의 합): <%= 1+2+3+4+5+6+7+8+9+10 %>
16  <hr/>
17  메소드 호출에 의한 결과 값: <%= cal.getTime() %>  // 오늘의 날짜와 현재 시각 출력
18  <hr/>
19  변수 i를 활용한 문자열과 연산의 결합: <%= "i+10=" + (i+10) %>  // i와 10을 더한 값을 문자열과 결합
20  </body>
21  </html>
```



2. 스크립팅 요소 (Scripting Element)

- 선언 (Declaration)
  - JSP 페이지의 스크립트릿이나 표현식에서 사용할 수 있는 멤버 변수 및 멤버 메소드를 작성할 때 사용
  - <%! ... %>
  - 멤버 변수 선언은 보통 하나의 문장으로 끝나기 때문에 반드시 뒤에 세미콜론(;) 필요

```
<%!
    멤버 변수 선언;
    멤버 메소드 선언
%>
```

```
<%!
    public 리턴타입 메소드이름(파라미터 리스트) {
        Java 코딩 문장;
        Java 코딩 문장;
        ...
        ...
        (return 값;)
    }
%>
```

2. 스크립팅 요소 (Scripting Element)

선언 (Declaration)

- 관련 예제

[예제 4.5] jspbook\ch04\decl.jsp

```
01  <%@ page contentType="text/html;charset=utf-8" %>
02  <%!  // 선언 블록 기호
03  String str = "선언 예제";   // 멤버 변수 str 선언
04  public double calculator(double num1, double num2, String operator) {  // 멤버 메소드 calculator 선언
05     double result = 0.0;
06
07     if (operator.equals("+")) {
08        result = num1 + num2;
09     } else if (operator.equals("-")) {
10        result = num1 - num2;
11     } else if (operator.equals("*")) {
12        result = num1 * num2;
13     } else if (operator.equals("/")) {
14        result = num1 / num2;
15     }
16       return result;
17  }
18  %>
```

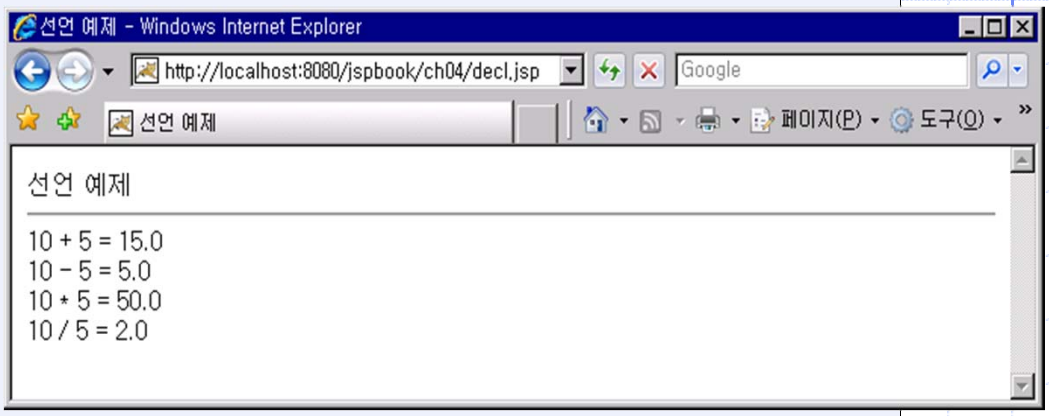
2. 스크립팅 요소 (Scripting Element)

선언 (Declaration)

관련 예제 (계속)

[예제 4.5] jspbook\ch04\decl.jsp

```
19  <html>
20  <head>
21  <title>선언 예제</title>
22  </head>
23  <body>
24  <%= str %>  // 선언된 str 출력
25  <hr/>
26  10 + 5 = <%= calculator(10, 5, "+") %><br/>
27  10 - 5 = <%= calculator(10, 5, "-") %><br/>
28  10 * 5 = <%= calculator(10, 5, "*") %><br/>
29  10 / 5 = <%= calculator(10, 5, "/") %>
30  </body>
31  </html>
```



선언된 calculator 메소드 호출에 의한 결과값 출력

2. 스크립팅 요소 (Scripting Element)

선언 (Declaration)

- 이전 JSP 예제가 변환된 .java 파일 내용

```
package org.apache.jsp.jspbook.ch04;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
public final class decl_jsp extends org.apache.jasper.runtime.HttpJspBase
implements org.apache.jasper.runtime.JspSourceDependent {
String str = "선언 예제";
public double calculator(double num1, double num2, String operator) {
   double result = 0.0;
   if (operator.equals("+")) {
      result = num1 + num2;
   } else if (operator.equals("-")) {
      result = num1 - num2;
   } else if (operator.equals("*")) {
      result = num1 * num2;
   } else if (operator.equals("/")) {
      result = num1 / num2;
   }
   return result;
}
```

JSP 선언 블록

2. 스크립팅 요소 (Scripting Element)

선언 (Declaration)

- 이전 JSP 예제가 변환된 .java 파일 내용 (계속)

```
private static final JspFactory _jspxFactory = JspFactory.getDefaultFactory();
private static java.util.List _jspx_dependants;
private javax.el.ExpressionFactory _el_expressionfactory;
private org.apache.AnnotationProcessor _jsp_annotationprocessor;
public Object getDependants() {
   return _jspx_dependants;
}
public void _jspInit() {
   _el_expressionfactory = _jspxFactory.getJspApplicationContext(getServletConfig().getServletContext()).getExpr
essionFactory();
   _jsp_annotationprocessor = (org.apache.AnnotationProcessor) getServletConfig().getServletContext().getAttri
bute(org.apache.AnnotationProcessor.class.getName());
}
public void _jspDestroy() {
}                                                    _jspService 멤버 메소드의 위치
public void _jspService(HttpServletRequest request, HttpServletResponse response)
throws java.io.IOException, ServletException {
... 중간 생략 ...
```

- JSP 선언 블록에서 선언된 요소들은 해당 클래스의 멤버 변수와 멤버 메소드로 정의된다.

3. 주석

- JSP 주석
  - JSP 소스 코드를 올바르게 이해하도록 하기 위하여 코드 자체에 설명을 달아 주는 것
  - 종류 1) JSP 주석

```
<%-- JSP 코드에 대한 설명 --%>
```

    - JSP 주석은 실행 시에 JSP 엔진 및 컨테이너가 아무런 처리 없이 바로 무시해 버린다.
    - 주의: JSP 주석이 중첩이 되는 일은 없어야 한다

```
<%-- JSP 코드에 대한 설명 <%-- 또 다른 설명 --%> --%>
```

3. 주석

JSP 주석

종류 2) Java 주석

- JSP의 스크립트릿, 표현식, 선언 블록 내부에는 Java언어를 사용하여 코딩을 하기 때문에 그러한 블록 내부에 사용가능한 주석

```
// Java 주석 첫 번째
/* Java 주석 두 번째 */
/** Java 주석 세 번째 */
```

종류 3) HTML 주석

- HTML 코드 자체에 주석을 다는 것
- 다른 주석들과 다르게 출력 결과에 포함되어지기 때문에 JSP 수행 결과 브라우저에서 '소스보기'를 하면 HTML 주석이 함께 보인다.

```
<!-- HTML 주석 -->
```

- 아래 코드에서 JSP 표현식("<%= userName %>")은 JSP 엔진에서 처리가 된 결과가 브라우저의 '소스보기'에서 보인다.

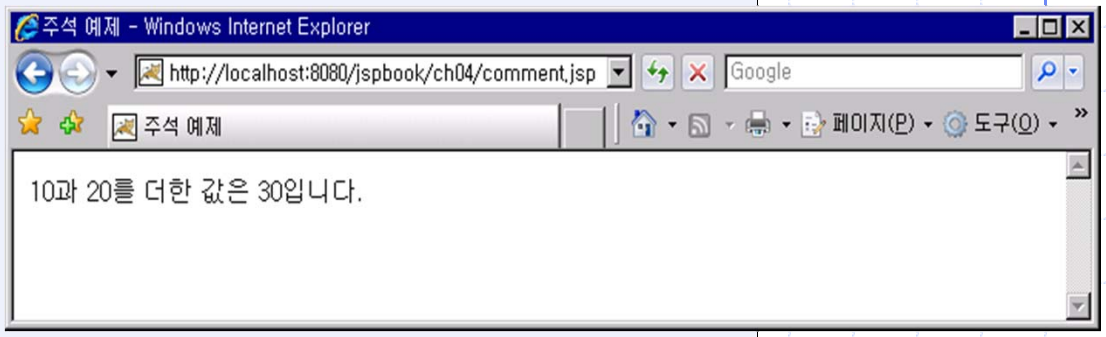
```
<!-- <%= userName %>이 작성하는 코드입니다. -->
```

3. 주석

JSP 주석

[예제 4.6] **jspbook\ch04\comment.jsp**

```
01  <%@ page contentType="text/html;charset=utf-8" %>
02  <%!
03  /** Java 주석 세 번째 것입니다 (plus: num1과 num2를 더한 값을 리턴한다.) */   // Java 주석
04  public int plus(int num1, int num2) {
05     return num1+num2;
06  }
07  %>
08  <html>
09  <head>
10  <title>주석 예제</title>
11  </head>
12  <!-- HTML 주석을 사용합니다. -->  // HML 주석
13  <body>
14  <%-- JSP 주석을 사용합니다. --%>  // JSP 주석
15  <%
16  // Java 주석 첫 번째 것입니다.  // Java 주석
17     int num1 = 10;
18     int num2 = 20;
19  /* Java 주석 두 번째 것입니다.    // Java 주석
20  num1과 num2를 더해서 result에 할당한다. */
21     int result = plus(num1, num2);
22  %>
23  <%= num1%>과 <%= num2%>를 더한 값은 <%= result%>입니다.
24  </body>
25  </html>
```



[브라우저의 소스보기]

```
<html>
<head>
<title>주석 예제</title>
</head>
<!-- HTML 주석을 사용합니다. -->
<body>
10과 20를 더한 값은 30입니다.
</body>
</html>
```