

## 제6장 JSP 기본 문법

1. 서버-사이드 언어인 JSP의 이해
2. 스크립트 태그의 이해
3. 디렉티브 태그의 이해

# 웹과 JSP 개념

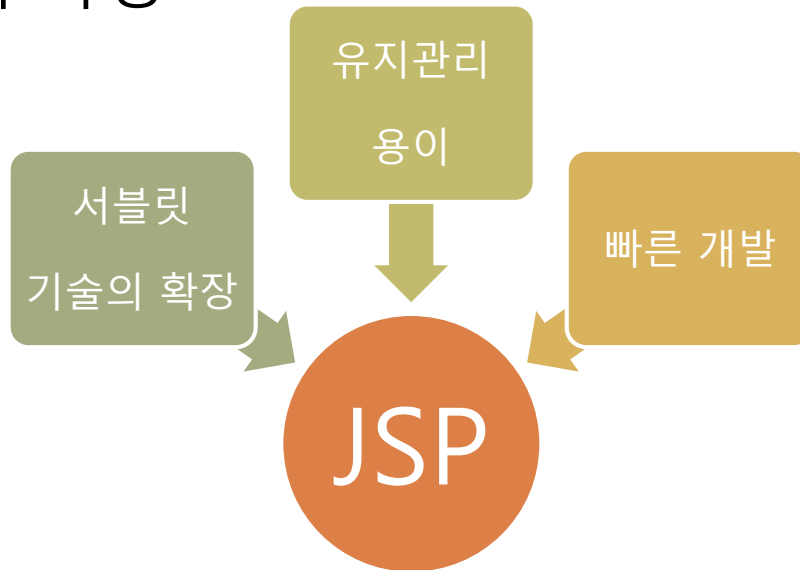
2

## □ JSP(Java Server Page)

- ▣ 자바를 기반으로 하는 서버측 웹 프로그래밍 언어
- ▣ 보안이 강한 스크립트 코드 방식의 개발 언어

: 프로그래머는 코드만 개발, 해당 문서의 요청이 있을 때 WAS가 컴파일을 수행하는 방식

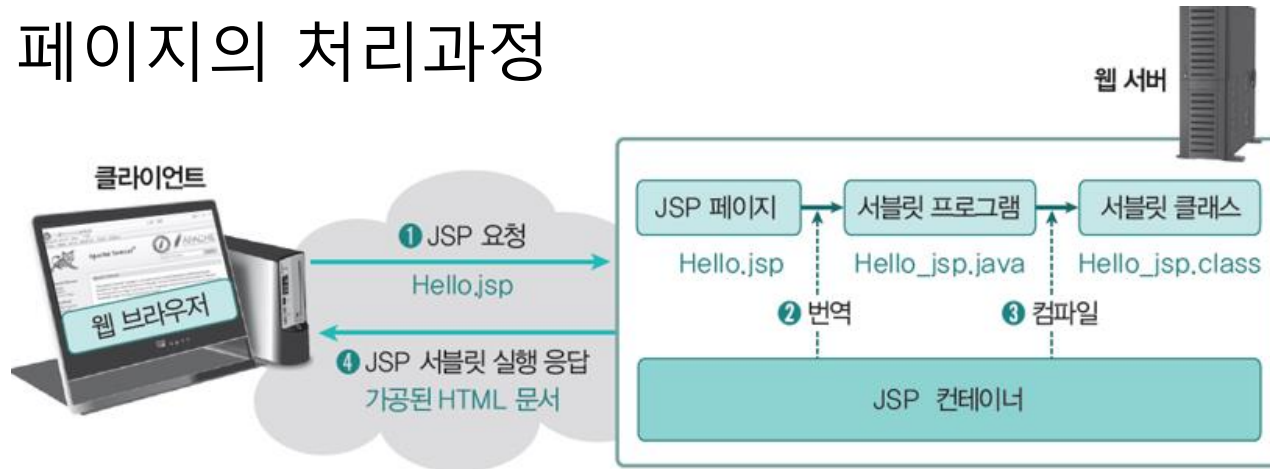
## □ JSP의 특징



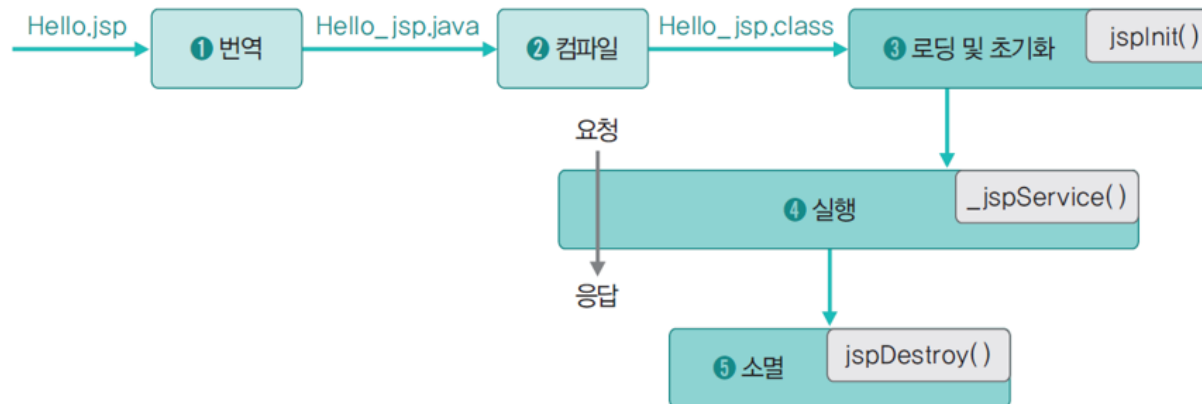
# 웹과 JSP 개념 - 계속

3

## □ JSP 페이지의 처리과정



## □ JSP 생명주기



# 스크립트 태그

4

## □ 스크립트 태그

- ▣ **<% ... %>** 사용
- ▣ JSP 페이지가 서블릿 프로그램에서 서블릿 클래스로 변환할 때 적용
  - JSP 컨테이너가 자바 코드가 삽입되어 있는 스크립트 태그를 처리하고 나머지는 HTML 코드나 일반 텍스트로 간주

## □ 스크립트 태그 종류

종류	태그 형식	사용 용도
지시어(directives)	<%@     %>	JSP 페이지의 속성을 지정
선언(declaration)	<%!     %>	소속변수 선언과 메소드 정의
표현식(expression)	<%=     %>	변수, 계산식, 함수 호출 결과를 문자열 형태로 출력
스트립트릿(scriptlet)	<%       %>	자바 코드를 기술
주석(comments)	<%--    --%>	JSP 페이지의 설명 추가

# 액션 태그와 커스텀 태그

5

## □ 액션 태그

- XML 스타일의 태그로 기술한 동작 기능을 수행하는 방식
- **<jsp:**지시자 속성="값" />

## □ 커스텀 태그

- 새로운 태그를 정의하여 이용하는 방법으로 **JSP** 만의 고유한 방식

종류	태그 형식	사용 용도
액션 태그 (action Tag)	<jsp:include page="test.jsp" />	현재 JSP 페이지에서 다른 페이지를 포함
	<jsp:forward page="test.jsp" />	현재 JSP 페이지의 제어를 다른 페이지에 전달
	<jsp:plugin type="applet" code="test" />	자바 애플릿을 플러그인
	<jsp:useBean id="login" class="LoginBean" />	자바빈을 사용
	<jsp:setProperty name="login" property="pass" />	자바빈의 속성을 지정하는 메소드를 호출
	<jsp:getProperty name="login" property="pass" />	자바빈의 속성을 반환하는 메소드를 호출
커스텀 태그 (Custom Tag)	<tag:printData dbname="mydb" table="memb" />	사용자가 직접 정의한 태그를 이용

# 스크립트릿

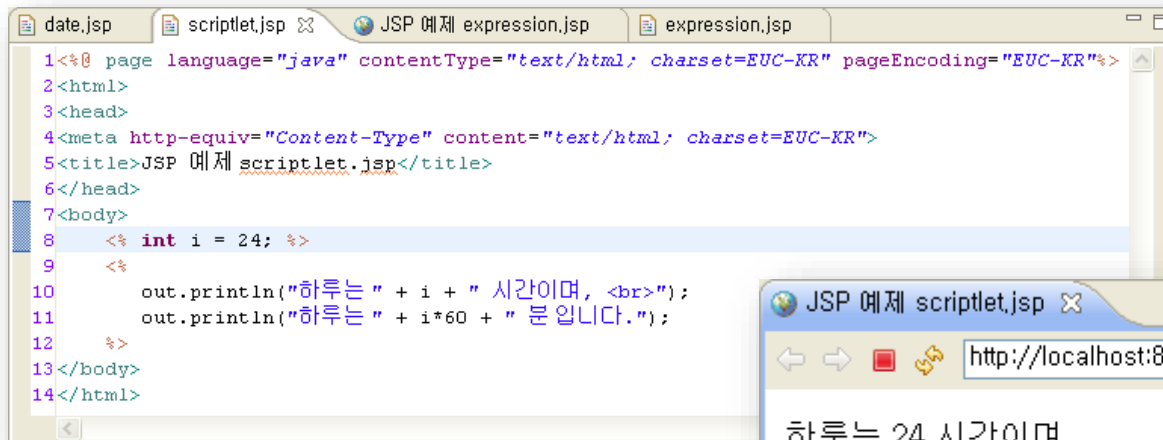
6

## □ 자바 코드 삽입

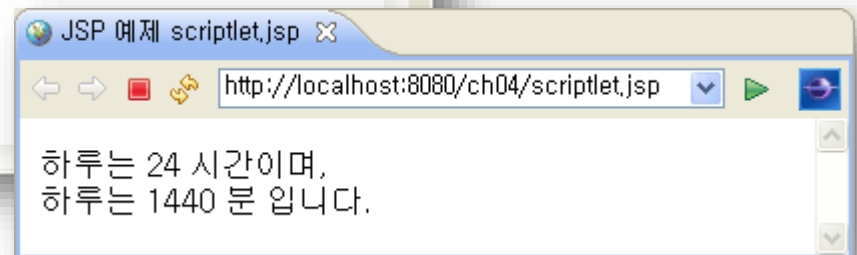
### ▣ 로직 부분 표현

- ▣ `<% code fragment ; %>` // 하나의 문장이 끝나면 ; 로 반드시 마감
- ▣ `<% out.print("스크립트릿 태그"); %>`

[실습1]



```
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 scriptlet.jsp</title>
6</head>
7<body>
8  <% int i = 24; %>
9  <%
10     out.println("하루는 " + i + " 시간이며, <br>");
11     out.println("하루는 " + i*60 + " 분 입니다.");
12  %>
13</body>
14</html>
```

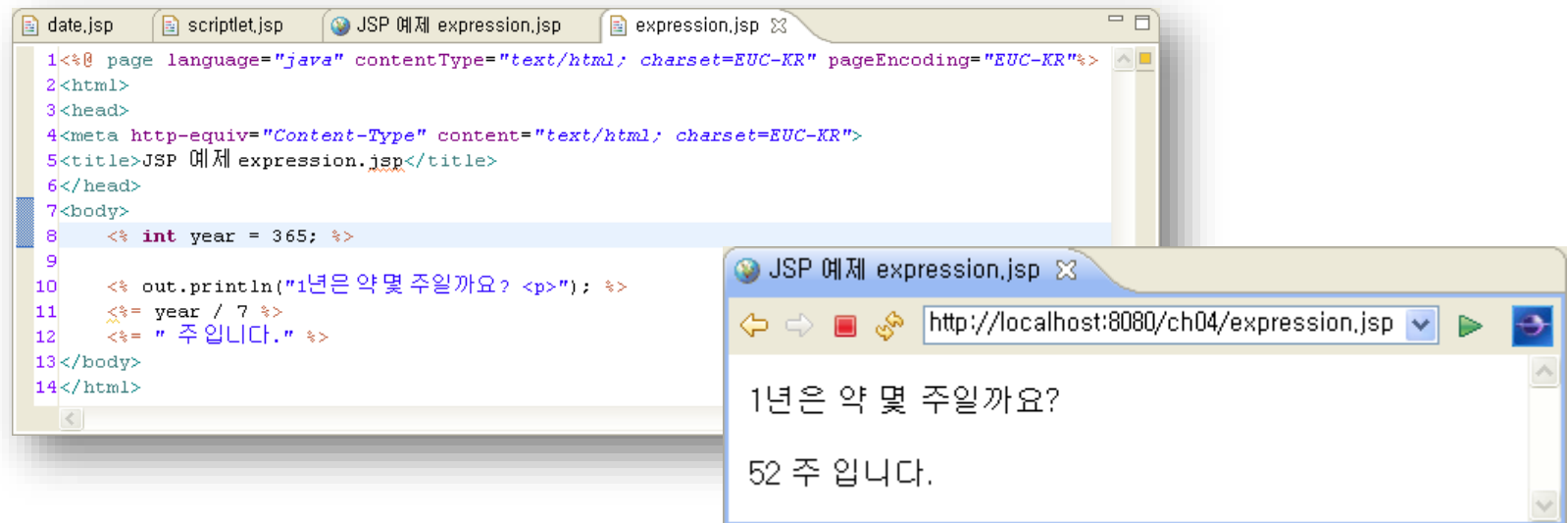


# 표현식

7

- 변수의 출력
- 웹 브라우저에서 출력할 부분 표현
- 숫자, 문자, Boolean 등의 기본 데이터 타입과 자바객체 사용 가능
- `<%= expression %>` // 문자 마감인 `;`를 사용할 수 없음

[실습2]



The screenshot shows a JSP development environment. On the left, a code editor displays the following JSP code:

```
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 expression.jsp</title>
6</head>
7<body>
8    <% int year = 365; %>
9
10    <% out.println("1년은 약 몇 주일까요? <p>"); %>
11    <%= year / 7 %>
12    <%= " 주 입니다." %>
13</body>
14</html>
```

On the right, a browser window titled "JSP 예제 expression.jsp" shows the rendered output at the URL `http://localhost:8080/ch04/expression.jsp`:

1년은 약 몇 주일까요?

52 주 입니다.

# JSP 소스의 서블릿 변환

8

- 이클립스에서 서블릿 위치
  - ▣ 톰캣 JSP 엔진
    - JSP 소스인 \*.jsp
      - 서블릿 소스 \*\_jsp.java로 자동 생성, 서블릿 클래스를 실행
  - ▣ 서블릿 파일의 위치
    - 서블릿 프로그램 실행 관련 클래스 파일이 있는 디렉토리
    - /WEB-INF/classes



# 실행 시 오류발생 : 컴파일 오류(1)

9

- 서블릿 변환 후 컴파일 오류
  - ▣ JSP 소스를 서블릿으로 변환 후 컴파일 시 발생하는 오류
    - 문법 오류(syntax error)
  - ▣ 이클립스 편집기에서도 소스 코딩 순간에 오류 표시

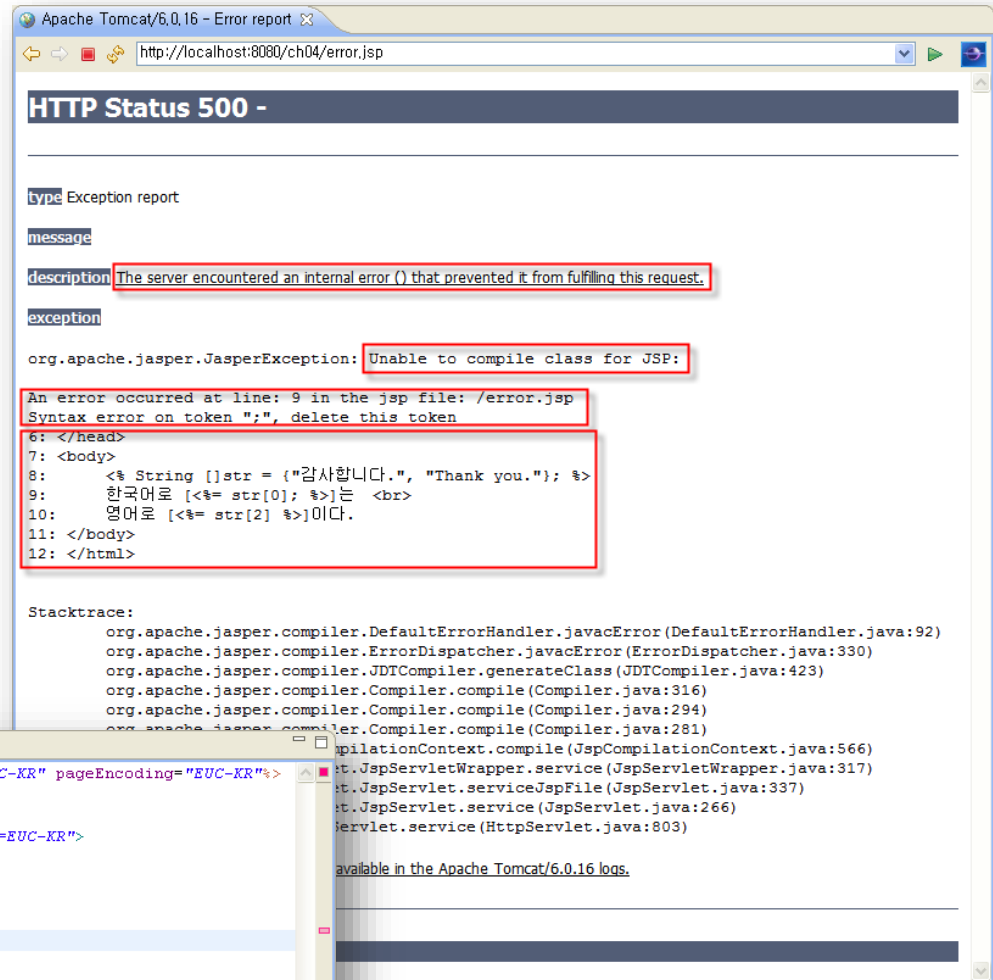
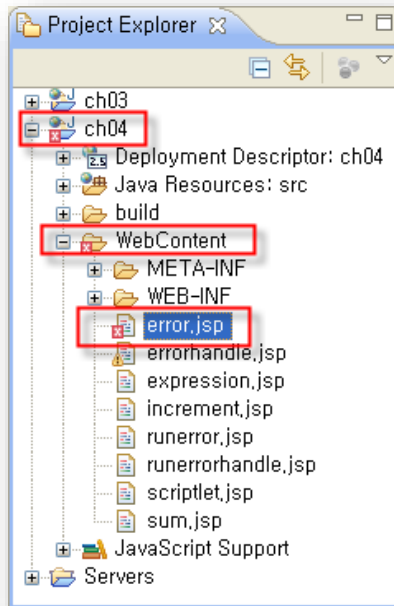


```
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 error.jsp</title>
6</head>
7<body>
8    <% String []str = {"감사합니다.", "Thank you."}; %>
9    한국어로 [<%= str[0]; %>]는 <br>
10   영어로 [<%= str[2] %>]이다.
11</body>
12</html>
```

# 실행 시 오류발생 : 컴파일 오류(2)

10

## 패키지 탐색기



# 서블릿 실행 시 오류

11

## □ 실행 시 오류

### ▣ 변환된 서블릿을 실행할 때 발생하는 오류

#### ■ `ArrayIndexOutOfBoundsException`

```
runerror.jsp
1<%@ page language="java" contentType="text/html; charset=EUC-KR" %>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 runerror.jsp</title>
6</head>
7<body>
8  <% String []str = {"감사합니다.", "Thank you."}; %>
9  한국어로 [<%= str[0] %>]는 <br>
10 영어로 [<%= str[2] %>]이다.
11</body>
12</html>
```

Apache Tomcat/6.0.16 - Error report

http://localhost:8080/ch04/runerror.jsp

### HTTP Status 500 -

Type: Exception report

Message:

Description: The server encountered an internal error () that prevented it from fulfilling this request.

Exception:

```
org.apache.jasper.JasperException: An exception occurred processing JSP page /runerror.jsp at line 10
7: <body>
8:   <% String []str = {"감사합니다.", "Thank you."}; %>
9:   한국어로 [<%= str[0] %>]는 <br>
10:  영어로 [<%= str[2] %>]이다.
11: </body>
12: </html>
```

Stacktrace:

```
org.apache.jasper.servlet.JspServletWrapper.handleJspException(JspServletWrapper.java:505)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:416)
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:337)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:266)
javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
```

Root cause:

```
java.lang.ArrayIndexOutOfBoundsException: 2
org.apache.jsp.runerror_jsp._jspService(runerror_jsp.java:67)
org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:374)
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:337)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:266)
javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
```

Note: The full stack trace of the root cause is available in the Apache Tomcat/6.0.16 logs.

Apache Tomcat/6.0.16

# 실행 오류 수정

12

## 배열 첨자 수정 후 실행

[실습3]

The screenshot displays an IDE with three windows. The top window, 'runerror.jsp.java', contains Java code for JSP page context initialization. The middle window, 'runerrorhandle.jsp', shows JSP code that defines a string array 'str' with two elements: '감사합니다.' and 'Thank you.'. The bottom window shows the browser output of the JSP page, displaying the Korean and English strings. A red box highlights line 67 in the JSP code, which prints the second element of the array.

```
runerror.jsp.java
38 JspWriter _jspx_out = null;
39 PageContext _jspx_page_context = null;

runerrorhandle.jsp
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 runerrorhandle.jsp</title>
6</head>
7<body>
8  <% String []str = {"감사합니다.", "Thank you."}; %>
9  한국어로 [<%= str[0] %>]는 <br>
10 영어로 [<%= str[1] %>]이다.
11</body>
12</html>

runerror.jsp.java
59 out.write("<body>\r\n");
60 out.write("\t");
61 String []str = {"한글", "Thank you."};
62 out.write("\r\n");
63 out.write("<body>대중 [");
64 out.print( str[0] );
65 out.write("<br>\r\n");
66 out.write("<body>공백");
67 out.print( str[2] );
68 out.write("<body>대중.\r\n");
69 out.write("</body>\r\n");
70 out.write("</html>");

JSP 예제 runerrorhandle.jsp
http://localhost:8080/ch04/runerrorhandle.jsp

한국어로 [감사합니다.]는
영어로 [Thank you.]이다.
```

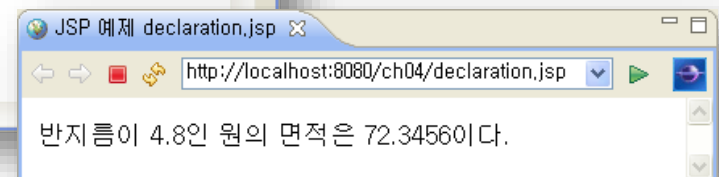
## □ 변수의 선언과 메소드 구현

- **<%! Declaration ; %>** // 문장 마감 기호인 ; 반드시 작성
- 선언 태그에서 선언되는 변수는 소속 변수(membered variables)
- 스크립트릿에서 선언되는 변수는 메소드 \_jspService() 내부에서 선언되는 지역 변수(local variables)

[실습4]

```
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 declaration.jsp</title>
6</head>
7<body>
8    <%! double radius = 4.8; %>
9    <%!
10        public double getArea(double r) {
11            return r * r * 3.14;
12        }
13    %>
14    반지름이 <%= radius %>인 원의 면적은 <%= getArea(radius) %>이다.
15</body>
16</html>
```

소속변수 또는 전역변수



# JSP의 지역변수와 소속변수

14

## □ 스크립트릿과 선언에서의 변수 선언

[실습5]

The image shows a JSP code editor and two browser windows. The code editor displays the following JSP code:

```
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 membervar.jsp</title>
6</head>
7<body>
8    <% int i = 0; %>
9    [지역변수] i = <%= ++i %>
10    <p>
11    [소속변수] memi = <%= ++memi %>
12    <% ! int memi = 0; %>
13</body>
14</html>
```

The first browser window shows the output after the first execution:

[지역변수] i = 1  
[소속변수] memi = 1

The second browser window shows the output after the second execution:

[지역변수] i = 1  
[소속변수] memi = 2

A speech bubble points to the second browser window with the text: "실행할 때마다 소속변수 memi는 증가하는 것을 알았다." (I realized that the member variable memi increases every time it is executed.)

# JSP의 지역변수와 소속변수

15

## □ 서블릿 소스

### ▣ membervar\_jsp.java

```
1 package org.apache.jsp;
2
3 import javax.servlet.*;
4 import javax.servlet.http.*;
5 import javax.servlet.jsp.*;
6
7 public final class membervar_jsp extends org.apache.jasper.runtime.HttpJspBase
8     implements org.apache.jasper.runtime.JspSourceDependent {
9
10     int memi = 0;
11     private static final JspFactory _jspxFactory = JspFactory.getDefaultFactory();
12
13     private static java.util.List _jspx_dependants;
14
15     private javax.el.ExpressionFactory _el_expressionfactory;
16     private org.apache.AnnotationProcessor _jsp_annotationprocessor;
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61     out.write("\t");
62     int i = 0;
63     out.write("\r\n");
64     out.write("\t[吏製跡畧] i = ");
65     out.print( ++i );
66     out.write("\r\n");
67     out.write("\t<p>\r\n");
68     out.write("\t[笑畧跡畧] memi = ");
69     out.print( ++memi );
70     out.write('\r');
71     out.write('\n');
72     out.write(' ');
73     out.write("\r\n");
74     out.write("</body>\r\n");
75     out.write("</html>");
76 } catch (Throwable t) {
77     if (!(t instanceof SkipPageException)){
78         out = _jspx_out;
```

## □ HTML 주석

- ▣ `<!-- ... -->`
- ▣ HTML 태그를 위한 주석으로 웹 브라우저의 [소스 보기]에서 HTML 내용과 함께 그 주석 내용이 보임.

## □ JSP 주석

- ▣ `<%-- ... --%>`
- ▣ JSP 서버 프로그램을 위한 주석으로 실행된 웹 브라우저의 [소스 보기]에서 표시되지 않음.

## □ HTML 주석 내부에서

- ▣ JSP의 스크립트릿 태그나 표현식 태그를 출력으로 이용가능  

```
<%  
    String str = "오후";  
%>  
<!-- 지금은 <%= str %>입니다. -->
```



# 주석 예제

17

[실습6]

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

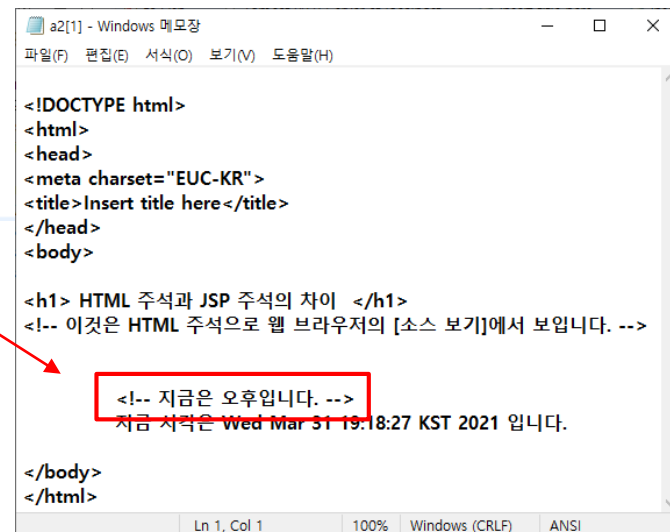
<h1> HTML 주석과 JSP 주석의 차이 </h1>
<!-- 이것은 HTML 주석으로 웹 브라우저의 [소스 보기]에서 보입니다. -->
<%-- 다음은 자바 코드의 스크립팅입니다. --%>
<%
    String str;
    if (java.util.Calendar.getInstance().get(java.util.Calendar.HOUR_OF_DAY) >= 12)
        str = "오후";
    else
        str = "오전";
%>
<!-- 지금은 <%= str %>입니다. -->
지금 시각은 <%= new java.util.Date() %> 입니다.

</body>
</html>
```

http://localhost:8080/2021web/a2.jsp

## HTML 주석과 JSP 주석의 차이

지금 시각은 Wed Mar 31 19:18:27 KST 2021 입니다.



# 디렉티브 태그(지시자)

18

## □ 디렉티브 형식

- 태그 형태를 이용하여 JSP 페이지에 대한 속성 또는 특별한 지시 사항을 지정하는 태그
- JSP 페이지와 관련된 정보를 JSP컨테이너에 지시하는 메시지
- `<%@ directives property="property-value" %>`
  - `directives`와 속성 `property` 모두 대소문자를 구분
  - 속성값은 반드시 “속성값”과 같이 큰따옴표를 이용

## □ JSP 지시자의 종류

- page, include, taglib 3가지

종류	형태	용도
page	<code>&lt;%@ page property="property-value" %&gt;</code>	JSP 페이지에 대한 속성 지정
include	<code>&lt;%@ include file="file-name" %&gt;</code>	태그 부분에 지정한 페이지를 정적으로 삽입
taglib	<code>&lt;%@ taglib uri="uri-value" prefix="pfx-value" %&gt;</code>	새로운 태그를 정의하여 이용

# page 디렉티브

19

## □ 기능

- JSP 컨테이너에서 JSP 페이지 대한 여러 속성과 값을 지정
- page 디렉티브는 language, contentType, pageEncoding 등의 속성을 지정
- 한번에 한 개의 속성 뿐만 아니라 서로 다른 여러 개의 속성을 지정
- `<%@ page 속성1="값1" [속성2="값" ...] %>`

```
<%@ page
[ language="java" ] [ extends="package.class" ]
[ import="{package.class | package.*} , ... " ]
[ session="true | false" ]
[ buffer="none | 8kb | sizekb" ] [ autoFlush="true | false" ]
[ isThreadSafe="true | false" ] [ info="text" ]
[ errorPage="relativeURL" ] [ isErrorPage="true | false" ]
[ contentType="{mimeType [ ; charset=characterSet ] |
  text/html ; charset=ISO-8859-1}" ]
[ pageEncoding="{characterSet | ISO-8859-1}" ]
[ isELIgnored="true | false" ]
%>
```

# language , contentType 속성

20

## □ language 속성

- JSP 페이지의 표현식, 선언, 스크립트릿에서 사용할 스크립트 언어의 종류를 지정하는 속성
- `<%@ page language="java" %>`

## □ contentType 속성

- JSP 페이지의 MIME(Multipurpose Internet Mail Extension) 유형(type)을 지정하는 속성
- 지정하지 않으면 “text/html”이 기본 값
  - MIME 유형
  - JSP 페이지 자료를 네트워크에서 주고 받을 때 서로 주고 받는 문서의 타입을 정의함으로써 이를 보내고 받는 시스템에서 원활하게 자료를 처리하려는 목적에서 나온 속성
- `<%@ page contentType="text/html" %>`
- `<%@ page contentType="text/html; charset=ISO-8859-1" %>`
- `<%@ page contentType="text/html; charset=EUC-KR" %>`

# pageEncoding, info 속성

21

## □ pageEncoding 속성

- ▣ JSP 페이지의 문자 인코딩 방식을 기술하는 속성
- ▣ 지정하지 않으면 기본 값이 [ISO-8859-1]
  - `<%@ page pageEncoding="ISO-8859-1" %>`
- ▣ 한글을 지원하기 위해서는 [EUC-KR]또는 [UTF-8]로 지정
  - `<%@ page pageEncoding="EUC-KR" %>`

## □ info 속성

- ▣ JSP 페이지 전체에 대한 설명이나 버전, 작성자, 작성 일자와 같은 정보를 문자열로 기술하는 부분
  - 길이에 제한이 없음.
  - `<%@ page info="JSP 페이지에 대한 설명이나 정보" %>`

# import 속성

22

## □ import 속성

- 자바의 `import` 문장을 대체하는 속성으로 이용할 클래스의 패키지를 지정하는 방법
- 자바의 `import` 문장과 달리, 필요하면 구분자 ,(coma)를 이용하여 여러 개의 클래스를 지정
- JSP 페이지에 여러 개의 `import` 페이지 속성도 기술 가능
- `<%@ page import="java.util.*" %>`
- `<%@ page import="java.util.Date, java.sql.*" %>`

## □ JSP 페이지에서 자동 import

- `[java.lang.*]`
- `[javax.servlet.*]`, `[javax.servlet.http.*]`, `[javax.servlet.jsp.*]`

# import 예제

23

[실습7]

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h2> page 지시자의 import 속성</h2>
    <%@ page import="java.util.Date" %>
    현재 시각 : <%= new Date().toLocaleString() %>

</body>
</html>
```

page 지시자의 import 속성

현재 시각 : 2021. 3. 31 오후 7:21:59

# isErrorPage, errorPage 속성

24

## □ isErrorPage 속성

- ▣ JSP 페이지가 오류를 처리하는 페이지인지를 **true** 또는 **false**로 지정하는 속성
  - 지정하지 않으면 기본값이 **false**이고, 필요하면 **true**로 지정
- ▣ `<%@ page isErrorPage="true" %>`
  - **isErrorPage**를 **true**로 지정하면 내장객체라 부르는 **exception** 변수를 사용하여 에러를 처리

## □ errorPage 속성

- ▣ JSP 페이지에서 발생한 오류를 처리하는 JSP 페이지를 기술하는 방법
- ▣ `<%@ page errorPage="exception.jsp" %>`
- ▣ 오류를 처리하는 전담 JSP 페이지를 지정
  - 이 시스템을 사용하는 사용자에게 일관성 있게 오류 처리를 해줄 수 있어 시스템의 신뢰성을 높일 수 있는 장점



# errorPage 예제

25

## □ errorpage.jsp, exception.jsp

[실습8]

```
errorpage.jsp
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 errorpage.jsp</title>
6</head>
7<body>
8  <%@ page errorPage="exception.jsp" %>
9  <% String []str = {"감사합니다.", "Thank you."}; %>
10  한국어로 [<%= str[0] %>]는 <br>
11  영어로 [<%= str[2] %>]이다.
12</body>
13</html>
```

```
exception.jsp
1<%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2<html>
3<head>
4<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5<title>JSP 예제 exception.jsp</title>
6</head>
7<body>
8  <%@ page isErrorPage="true" %>
9  <h2> 처리 중 문제 발생 </h2>
10  빠른 시일 내에 복구하도록 하겠습니다. <p>
11  exception.getMessage() : <%= exception.getMessage() %> <p>
12  exception.toString() : <%= exception.toString() %>
13</body>
14</html>
```

### 처리 중 문제 발생

빠른 시일 내에 복구하도록 하겠습니다.

exception.getMessage() : 2

exception.toString() : java.lang.ArrayIndexOutOfBoundsException: 2

# isThreadSafe, isELIgnored 속성

26

## □ isThreadSafe 속성

- isThreadSafe 속성은 동시 사용자 접속 처리에 대한 지정 방법
- true 또는 false로 지정, true가 기본 값
- `<%@ page isThreadSafe="false" %>`

## □ isELIgnored 속성

- 표현 언어인 EL(Expression Language)의 사용 여부를 지정하는 방법
- 지정하지 않으면 false가 기본 값
  - 표현 언어를 사용한다는 의미
- 만일 표현 언어를 사용하지 않으려면 값을 true로 지정
- `<%@ page isELIgnored="true" %>`

# buffer 속성

27

## □ buffer 속성

- ▣ JSP 페이지의 출력 버퍼링 메모리를 크기를 지정하는 방법
- ▣ 지정하지 않으면 8kb가 기본 값
- ▣ buffer 속성 값은 none 또는 16kb와 같이 다른 크기의 값으로 지정할 수 있다.
- ▣ `<%@ page buffer="16kb" %>`

## □ 버퍼링

- ▣ 일반적으로 입력이나 출력에 이용하는 방식으로 프로세스의 처리 속도보다 입출력의 속도가 느리기 때문에 어느 정도 자료를 모아서 입출력을 처리하는 방법
- ▣ 양동이에 어느 정도의 물이 차야 물을 쏟아내는 양동이와 같이 버퍼링 메모리에 지정한 크기만큼의 자료가 쌓여야 출력을 하는 방식
- ▣ `<%@ page buffer="none" %>`
  - buffer 속성 값이 none이면 버퍼링을 하지 않겠다는 의미로 출력 자료가 버퍼를 거치지 않고 바로 웹 브라우저에 출력

# autoFlush 속성

28

## □ autoFlush 속성

- 버퍼가 모두 찼을 때 자동으로 출력하는지를 지정하는 방법
- 지정하지 않으면 **true**가 기본 값으로,
  - 버퍼 크기의 자료가 모두 찼을 때 자동으로 웹 브라우저에 출력한다는 의미
- `<%@ page autoFlush="false" %>`
  - 만일 **autoFlush** 속성을 **false**로 지정하면 버퍼 크기 만큼 차기 전, 중간 중간에 수동으로 직접 버퍼를 비워야 출력이 가능
  - 버퍼 크기의 자료가 모두 찼을 경우, 오버플로우(**overflow**) 예외가 발생

## □ 주의점

- 다음과 같이 **autoFlush** 값이 **false**이면 수동으로 버퍼링을 해야 하는데, **buffer** 값을 **none**으로 지정하면 버퍼링을 하지 않겠다는 것으로 잘못된 지정 방법
  - `<%-- 다음은 잘못된 page 버퍼 지정 방법이다. --%>`
  - `<%@ page buffer="none" %>`
  - `<%@ page autoFlush="false" %>`

# session 속성

29

## □ session 속성

- JSP 페이지에서 세션을 사용하는지의 여부를 지정하는 방법
- 지정하지 않으면 **true**가 기본 값으로 세션을 이용 가능
- `<%@ page session="false" %>`

## □ 세션(session)

- 웹 브라우저의 사용자를 구분하는 단위로 사용자 별로 웹 서버에 필요한 정보를 임시로 저장하는 방법
- 세션 예
  - 어느 사이트에 접속했을 때 한 번 로그인 한 후 어느 정도 시간이 지나면 세션이 끊어져 더 이상 사용할 수 없다거나
  - 장바구니에 새로운 상품을 이전 목록에 추가

# page 디렉티브 태그 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.	

# include 디렉티브 태그

31

## □ file 속성

- include 지시자는 태그를 기술한 부분에 지정한 파일을 삽입하는 기능
- 유일한 속성으로 삽입 파일을 지정하는 file
- include 디렉티브 태그는 JSP 페이지 어디에서든 선언 가능
- `<%@ include file="file_name" %>`

[실습9]

```
include.jsp
1 <%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5 <title>JSP 예제 include.jsp </title>
6 </head>
7 <body>
8     <%@ include file="header.jsp" %>
9     <hr> <p>
10    include 지시자 : <%=> include file="file_name" %> <p>
11    <hr> <p>
12    <%@ include file="footer.html" %>
13 </body>
14 </html>
15
```

# include 예제

32

## □ header.jsp, footer.html

[실습9]

```
header.jsp
1 <%@ page language="java" contentType="text/html; charset=EUC-KR" pageEncoding="EUC-KR"%>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
5 <title>JSP 예제</title>
6 </head>
7 <body>
8 <h2><font color=blue>JSP 지시자 예제</font></h2>
9 </body>
10 </html>
```

```
*footer.html
1 <%@ page contentType="text/html; charset=EUC-KR" %>
2 <center>
3 <font color=red>소프트웨어 융합 대학</font>
4 </center>
5
```

### JSP 지시자 예제

include 지시자 : <%@ include file="fine\_name" %>

소프트웨어 융합 대학



# 스몰과제(6)

33

- 강의 동영상 자료에 있는 실습 예제(총 9개)를 모두 프로그램 한 후 소스와 출력 결과를 편집하여 제출하세요.

- 
- 제출방법 : ppt파일에 소스와 출력 결과 편집 후 제출
  - 제출마감일 : 2021년 04월12일 월요일 오후 3시 까지
  - 제출장소 : 한림 스마트 LEAD(<https://smartlead.hallym.ac.kr>) 해당 과목 [과제]란에 제출하시면 됩니다. (제출시 제목란에 "여러분의 학번이름"을 쓰시고, 파일을 전송하시면 됩니다.)