

4

JSP 기본 요소



이 장에서 다룰 내용

1

JSP 주석

2

JSP 지시어(Directive)

3

JSP 스크립트 요소



JSP 주석

□언어별 주석의 형태

■ HTML 주석

```
<!-- HTML 주석입니다. -->
```

■ JSP 주석

```
<%-- JSP 주석입니다. %-->
```

-- %-->

■ 자바 스타일 주석

```
<%  
/* 여러 줄을 사용 할 때의  
자바 주석입니다. */  
%>
```

```
<%  
// 한 줄을 사용할 때의 자바 주석입니다.  
%>
```



JSP 지시어(Directive)

□ JSP 지시어(Directive)

- JSP 파일 내에서 JSP를 실행할 컨테이너에서 해당 페이지를 어떻게 처리할 것인가에 대한 설정 정보들을 지정해주는 데 사용
- 지시어 종류
 - Page 지시어, include 지시어, taglib 지시어

```
<%@ ... %>
```

□ page 지시어

- JSP 페이지에 대한 속성을 지정하는 지시어

```
<%@ page 속성1="값1" 속성2="값2" 속성3="값3" ... %>
```



JSP 지시어(Directive)

속성	사용법	기본값	설명
language	✓ language="java"	java	스크립트 요소에서 사용할 언어 설정
extends	extends="클래스명"	없음	상속받을 클래스를 설정
import	✓ import="패키지/클래스명"	없음	import 할 패키지/클래스 설정
session	session="true"	true	HttpSession을 사용 여부를 설정
buffer	buffer="16kb"	8kb	JSP 페이지의 출력 버퍼의 크기를 설정
autoFlush	autoFlush="true"	true	출력 버퍼가 다 찼을 경우 처리 방법을 설정
isThreadSafe	isThreadSafe="true"	true	다중 스레드의 동시 실행 여부를 설정
Info	info="페이지 설명"	없음	페이지 설명
errorPage	errorPage="에러페이지.jsp"	없음	에러 페이지로 사용할 페이지를 지정
contentType	✓ contentType="text/html"	text/ html; charset=ISO-8859-1	JSP 페이지가 생성할 문서의 타입을 지정
isErrorPage	isErrorPage="false"	false	현재 페이지를 에러 페이지로 지정
pageEncoding	✓ pageEncoding="euc-kr"	ISO-8859-1	현재 페이지의 문자 인코딩 타입 설정



JSP 지시어(Directive)

■ language 속성

- JSP 파일 내에서 사용될 스크립트 언어를 지정하는 속성이다. 특별히 지정하지 않을 경우 JSP 컨테이너가 지원하는 기본 언어인 자바가 language 기본 속성으로 지정된다.

```
<%@ page language="java" %>
```

■ extends 속성

- 서블릿 클래스로 변환되는 과정에서 상속할 부모 클래스를 지정하는 속성이 extends 속성이다. 기본적으로 이러한 상속은 JSP 컨테이너에서 자체적으로 알아서 처리하기 때문에 개발자가 특별히 지정하는 경우는 거의 없다.
- JSP 컨테이너에서 상속받게 되는 클래스인 javax.servlet.jsp.HttpJspPage 또는 javax.servlet.jsp.JspBase 인터페이스를 구현한 클래스를 상속받아야 한다.


```
<%@ page extends="javax.servlet.jsp.HttpJspBase" %>
```

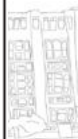


JSP 지시어(Directive)

■ import 속성

- JSP 페이지 내에서 사용할 패키지 또는 클래스를 import 하는 데 사용된다.

importTest.jsp		 Chapter4Wex1WimportTest.jsp	
1	<%@ page language="java" contentType="text/html; charset=EUC-KR"		
2	pageEncoding="EUC-KR"%>		
3	<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/		
4	loose.dtd">		
5			
6	<%@page import="java.text.SimpleDateFormat"%>		
7	<%@page import="java.util.*"%>		
8	<html>		
9	<head>		
10	<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">		
11	<title>import 속성 테스트</title>		
12	</head>		
13	<body>		
14	<h1> 현재 시간은 <%=new SimpleDateFormat().format(new Date()) %>입니다.</h1>		
15	</body>		
16	</html>		



JSP 지시어(Directive)

■ session 속성

- Stateless 특성으로 인해 커넥션을 매번 새로 생성하기 때문에 동일한 클라이언트에 대해 정보를 유지할 필요가 있는 경우(예를 들어 로그인 상태를 유지할 필요가 있는 경우 등) 사용

```
<%@ page session="true" %>
```

71.7%
2.6%

■ buffer 속성

- JSP 페이지가 서블릿으로 변환되어 응답을 생성해 클라이언트에 전송할 때 사용되는 버퍼의 크기를 지정하는 속성이다.
- 기본적으로는 “8kb”로 설정되어 있으며 만일 “none”으로 설정할 경우 응답을 생성할 때 버퍼를 사용하지 않고 바로 바로 클라이언트에 응답을 전송한다.

```
<%@ page buffer="16kb" %>
```



JSP 지시어(Directive)

■ autoFlush 속성

- buffer 속성에서 지정한 버퍼가 다 채워졌을 경우 이를 어떻게 처리할 것인지 JSP컨테이너에게 알려주는 속성이다. 기본값은 “true” 로, 버퍼가 다 채워지면 클라이언트에 전송하고 버퍼를 비운다. autoFlush 속성을 “false” 로 지정하면 버퍼가 다 채워졌을 경우에는 버퍼를 전송하는 것이 아니라 Exception을 발생시킨다. 만일 buffer 속성을 “none” 으로 지정했을 경우에는 autoFlush 속성을 “false” 로 지정하는 것은 불가능하다.

```
<%@ page autoFlush="true" %>
```

■ isThreadSafe 속성

- 기본적으로는 다중 스레드의 동시 실행을 허용하는 “^{true}” 로 지정되어 있다. isThreadSafe 속성을 “false” 로 지정하면 JSP 파일이 실행되더라도 하나의 스레드에 의한 service() 메소드 실행이 완전히 끝날 때까지 다른 스레드는 기다리게 된다. 따라서 해당 페이지에 의해 공유되는 데이터에 대한 사용자의 동시 접근을 막을 수 있게 된다.

```
<%@ page isThreadSafe="true" %>
```



JSP 지시어(Directive)

■ info 속성

- 해당 페이지 전체에 대한 주석의 역할을 하는 속성으로 개발자가 해당 페이지의 기능이나 특성을 설명해 놓을 때 활용된다.

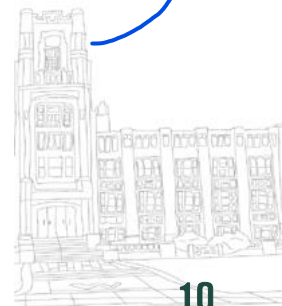
```
<%@ page info="이 페이지는 이클립스로 만들어진 페이지입니다." %>
```

■ errorPage 속성

- JSP 페이지를 처리하는 과정에서 예외(Exception)가 발생했을 때 해당 페이지에서 직접 처리하지 않고 에러 페이지를 지정해 그 에러 페이지에서 예외를 처리하도록 할 때 사용하는 속성이다.

```
<%@ page errorPage="error/error.jsp" %>
```

- ~~buffer 속성이 “none” 으로 지정된 경우와 autoFlush가 “true” 로 지정된 경우는 에러페이지를 사용할 수 없다.~~



JSP 지시어(Directive)

■ contentType 속성

- JSP 페이지가 전송할 응답의 형태가 어떠한 MIME 형식으로 되어 있는지를 지정하는 속성이다.

```
<%@ page contentType="text/html" %>
```

- 문자 셋(Character Set)의 인코딩 방식을 “;charset=” 을 사용해 추가할 수 있다.

```
<%@ page contentType="text/html; charset=euc-kr" %>
```

■ isErrorPage 속성

- 현재 페이지가 다른 페이지에서 발생한 예외(Exception)를 처리할 페이지일 때 지정하는 속성이다.
- 기본값은 당연히 “false” 이고 특정 JSP 페이지의 예외를 처리할 JSP 에러 페이지를 만들었다면 다음처럼 그 에러 페이지의 isErrorPage 속성을 “true” 로 지정하여 사용한다.

```
<%@ page isErrorPage="true" %>
```

JSP 지시어(Directive)

■ pageEncoding 속성

- JSP 페이지의 문자 셋 인코딩 방식을 지정할 때 사용하는 속성이다.
- 기본값은 “ISO-8859-1”로 이는 북유럽 문자 셋 인코딩 방식이므로 한글을 처리할 수가 없다. 따라서 속성 값을 다음처럼 한글이 호환되는 문자 셋 인코딩 방식인 “euc-kr”로 지정하지 않을 경우 한글이 모두 깨져서 출력되게 된다.

```
<%@ page pageEncoding="euc-kr" %>
```

- 보통은 pageEncoding 속성을 사용하지 않고 지시자 영역에서 contentType 속성을 지정할 때 한 것처럼 한번에 지정하여 사용한다.

```
<%@ page contentType="text/html; charset=euc-kr" %>
```

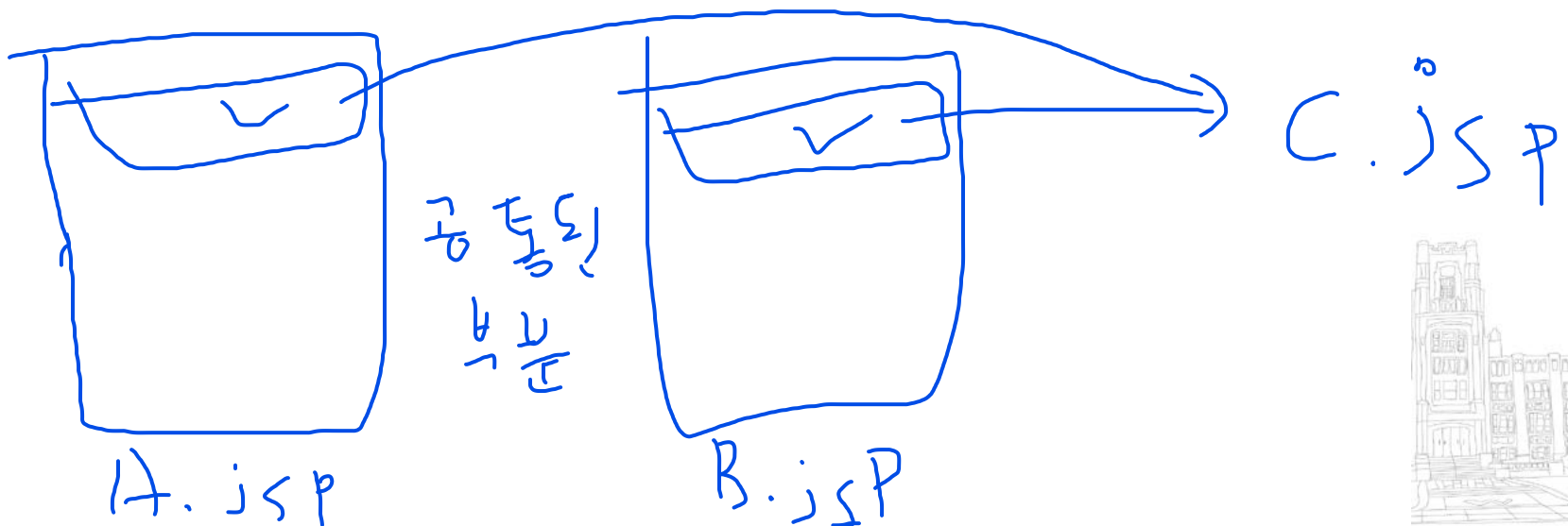


JSP 지시어(Directive)

□ include 지시어

- 특정한 JSP 파일 또는 HTML 파일을 해당 JSP 페이지에 삽입할 수 있도록 하는 기능을 제공하는 지시어
- 여러 JSP 페이지에서 공통되는 부분이 많을 때 이러한 공통되는 부분을 따로 파일로 만들어 include 지시어로 삽입

```
<%@ include file="header.jsp" %>
```



JSP 지시어(Directive)

■ 예제

includeTest.jsp



Chapter4Wex2WinincludeTest.jsp

```
1  <%@ page language="java" contentType="text/html; charset=EUC-KR"
2      pageEncoding="EUC-KR"%>
3  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
4  loose.dtd">
5  <%@ include file="header.jsp"%>
6  <html>
7  <head>
8  <meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
9  <title>include 테스트</title>
10 </head>
11 <body>
12 <h1> includeTest.jsp 파일입니다.</h1>
13 <%@ include file="footer.jsp" %>
14 </body>
15 </html>
```



JSP 지시어(Directive)

header.jsp



Chapter4Wex2Wheader.jsp

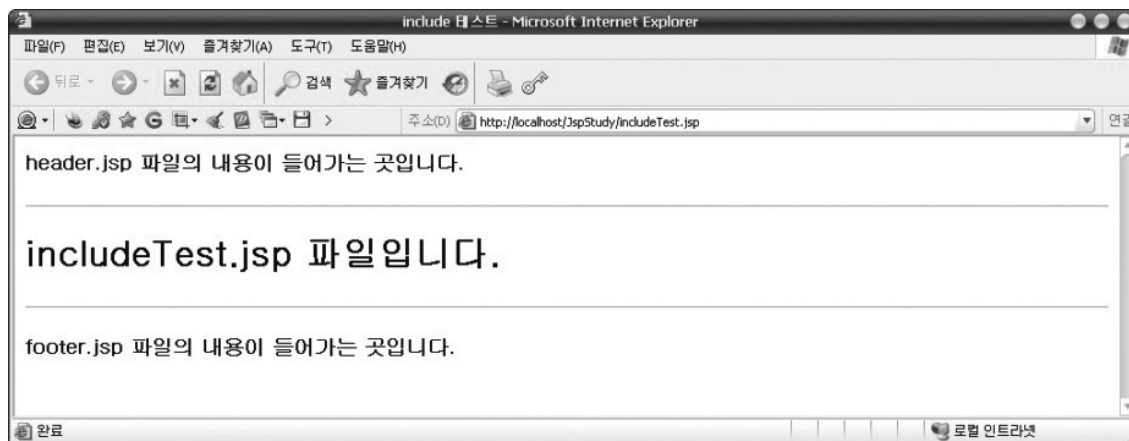
```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3 <h3>header.jsp 파일의 내용이 들어가는 곳입니다.</h3>
4 <hr>
```

footer.jsp



Chapter4Wex2Wfooter.jsp

```
1 <%@ page language="java" contentType="text/html; charset=EUC-KR"
2   pageEncoding="EUC-KR"%>
3 <hr>
4 <h3>footer.jsp 파일의 내용이 들어가는 곳입니다.</h3>
```



JSP 지시어(Directive)

□taglib 지시어

나중에

- JSTL(JSP Standard Tag Library)이나 커스텀 태그 등 태그 라이브러리를 JSP에서 사용할 때 접두사를 지정하기 위해 사용
- uri 속성과 prefix 속성의 두 가지 속성으로 구성
 - uri 속성 - 태그 라이브러리에서 정의한 태그와 속성 정보를 저장한 TLD(Tag Library Descriptor) 파일이 존재하는 위치를 지정
 - prefix 속성 - 사용할 커스텀 태그의 네임스페이스(Name Space)를 지정

```
<%@ taglib uri="http://taglib.com/sampleURI" prefix="samplePrefix" %>
```

- uri에 존재하는 TLD 파일에 정의된 태그들을 prefix로 지정한 네임스페이스를 통해 다음과 같이 사용

```
<samplePrefix: table col="2" row="2" border="1" >
```

태그 라이브러리를 이용한 테이블입니다.

```
</samplePrefix: table>
```



JSP 스크립트 요소

□ JSP 스크립트 요소

- JSP 페이지 내에 자바의 코드를 삽입하기 위해 사용
- 선언문(Declarations), 스크립틀릿(Scriptlets), 표현식(Expressions)의 3가지로 구분

□ 선언문(Declarations)

- <%!...%> 태그를 사용 *필수지점*

```
<%!  
private String str="JSP 스터디";  
public String checkStr(){  
    if(str==null) return "no";  
    else return "ok";  
}  
%>
```

```
<%! private String str="JSP 스터디"; %>  
<%!  
public String checkStr(){  
    if(str==null) return "no";  
    else return "ok";  
}  
%>
```

```
<%!  
public String checkStr(){  
    if(str==null) return "no";  
    else return "ok";  
}  
%>  
<%! private String str="JSP 스터디"; %>
```



JSP 스크립트 요소

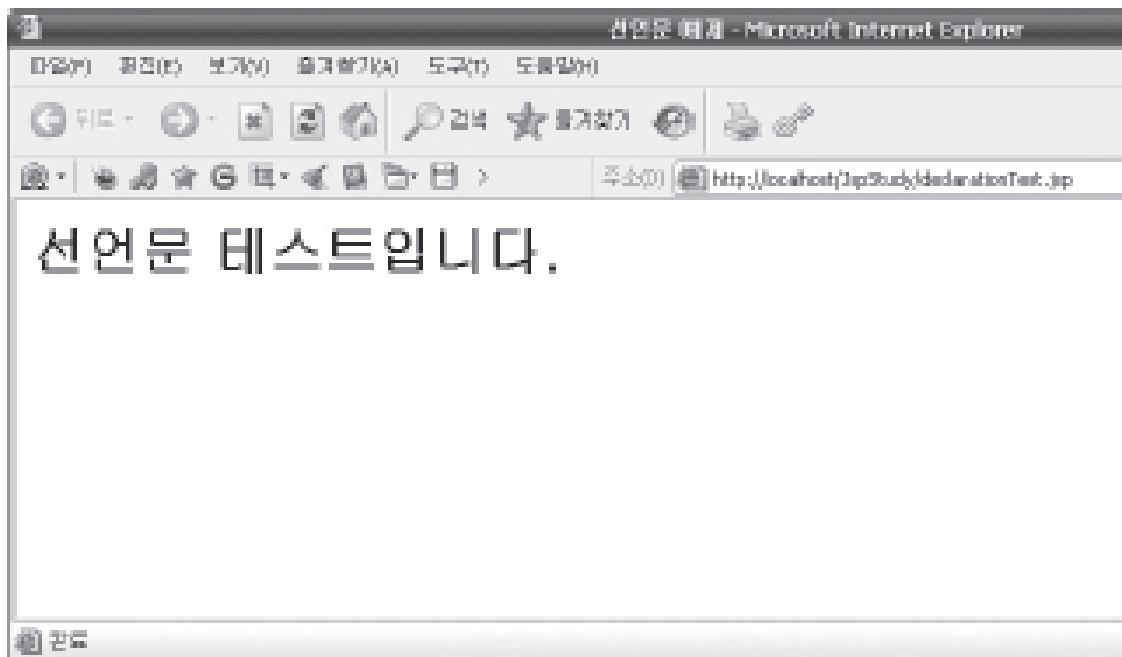
■ 예제

declarationTest.jsp



Chapter4\Wex3\declarationTest.jsp

```
1  <%@ page language="java" contentType="text/html; charset=EUC-KR"
2      pageEncoding="EUC-KR"%>
3
4  <h1><%=getStr() %> </h1>
5  <%!
6      private String getStr(){
7          str+="테스트입니다.";
8          return str;
9      }
10     private String str="선언문 ";
11     %>
```



JSP 스크립트 요소

□ 스크립틀릿(Scriptlet)

- HTML 코드로 된 부분은 일반 HTML 파일처럼 그대로 사용
- 자바 코드로 이루어진 로직 부분은 <%...%>로 표현되는 스크립틀릿 태그를 사용

```
<% 문장1; %>
```

```
<%
```

```
문장2; 문장3; 문장4; ...
```

```
%>
```

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"  
    pageEncoding="EUC-KR"%>
```

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

```
<%
```

```
Calendar c=Calendar.getInstance();
```

```
int hour=c.get(Calendar.HOUR_OF_DAY);
```

```
int minute=c.get(Calendar.MINUTE);
```

```
int second=c.get(Calendar.SECOND);
```

```
%>
```

```
<html>
```

```
<head>
```

```
<title>Scriptlet Test</title>
```

```
</head>
```

```
<body>
```

```
<h1>현재시간은 <%=hour %>시 <%=minute %>분 <%=second %>초 입니다.</h1>
```

```
</body>
```

```
</html>
```

표현식(Expression)

□ 표현식(Expression)

- 선언문 또는 스크립틀릿 태그에서 선언된 변수나 메소드의 리턴값을 스크립틀릿 태그 외부에서 출력하기 위해 사용되는 방법

<%=변수 %>

<%=리턴값이 있는 메소드 %>

<%=수식(변수 또는 리턴값이 있는 메소드를 포함할 수 있음) %>

expressionTest.jsp



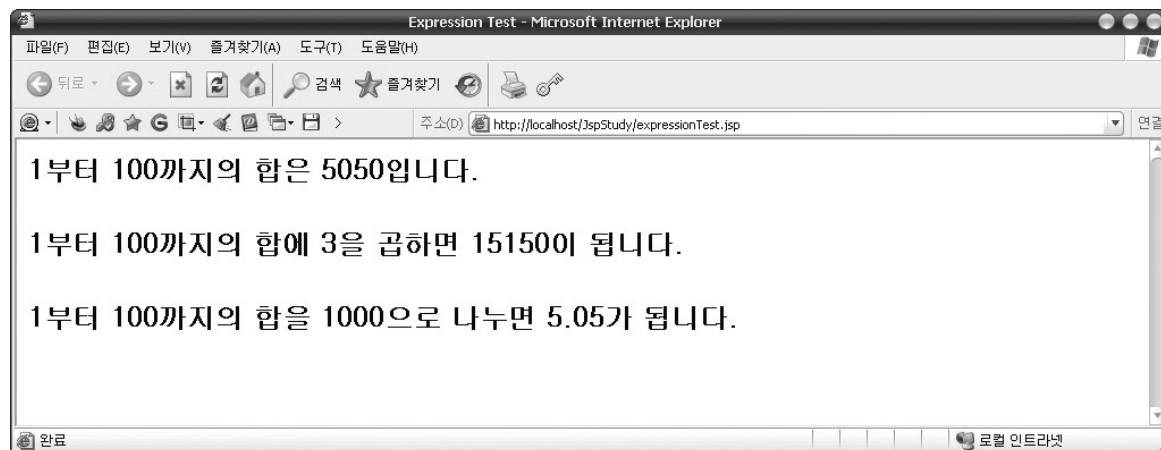
Chapter4\Wex6\WexpressionTest.jsp

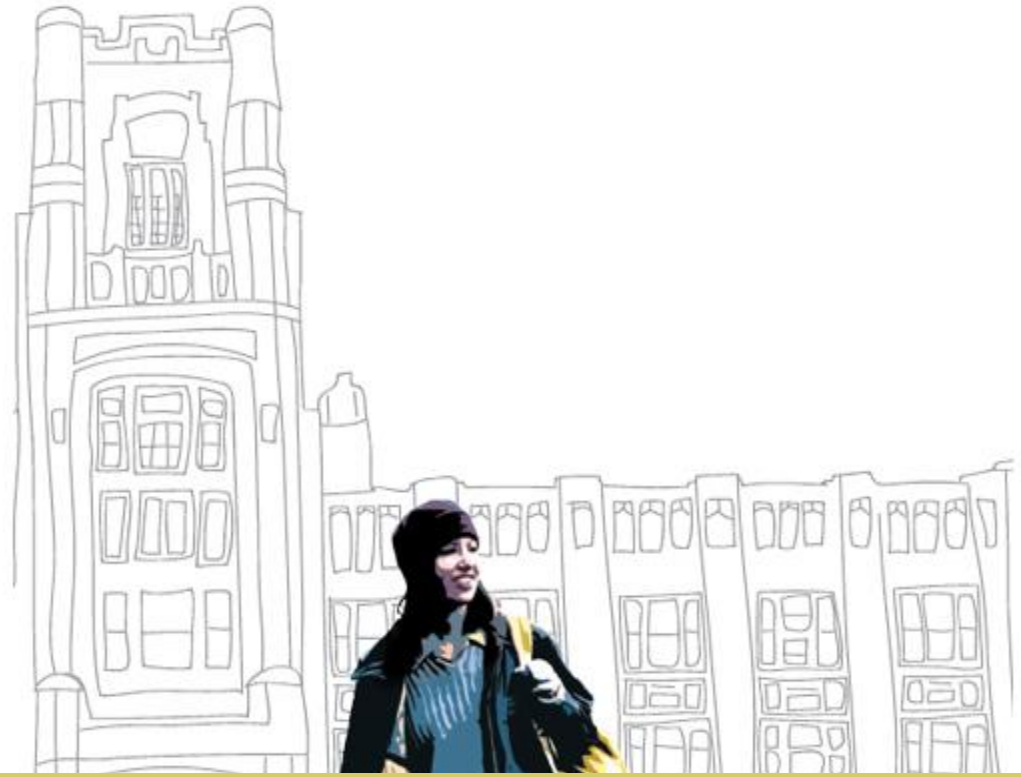
```
1  <%@ page language="java" contentType="text/html; charset=EUC-KR"
2      pageEncoding="EUC-KR"%>
3  <%!
4      public int sum(){
5          int total=0;
6          for(int i=1;i<=100;i++){
7              total+=i;
8          }
9          return total;
10     }
11     %>
12     <%
13     String str="1부터 100까지의 합";
14     %>
```



표현식(Expression)

```
15 <html>
16 <head>
17 <title>Expression Test</title>
18 </head>
19 <body>
20 <h2><%=str %>은 <b><%=sum() %></b>입니다.</h2>
21 <br>
22 <h2><%=str %>에 3을 곱하면 <b><%=sum()*3 %></b>이 됩니다.</h2>
23 <br>
24 <h2><%=str %>을 1000으로 나누면 <b><%=sum()/1000. %></b>가 됩니다.</h2>
25 </body>
26 </html>
```





Thank You

