

2장. JSP 프로그래밍 기초

컴퓨터정보과
김진숙

이번 장에서 공부할 것

- JSP 프로그램을 위한 기본적인 사항들
- 학습내용
 - JSP 프로그램의 기본 형태
 - JSP 프로그램의 저장 위치와 실행 방법
 - 주석(Comment)
 - 화면 출력

이 단원이 끝나면 알게 되는 것

1. 스크립틀릿(JSP 프로그램 안에 있는 자바 코드)의 시작과 끝은 어떻게 표시하는지, 또 페이지 지시자의 시작과 끝은 어떻게 표시하는지 적어보시오.
2. 톰캣의 웹 애플리케이션 베이스 폴더(webapps)에 test라는 폴더를 만들고, 2-1.jsp이라는 이름의 프로그램 파일을 이 폴더에 넣었다고 할 때, 이 파일을 실행시키기 위해 웹 브라우저의 주소창에 써넣어야 하는 URL을 적어보시오.
3. 주석에 대한 다음 질문에 답해보시오.
 - ① 한 줄 주석의 시작 표시는? ()
 - ② 여러 줄 주석의 시작과 끝 표시는? ()
 - ③ JSP 주석의 시작과 끝 표시는? ()
 - ④ HTML 주석의 시작과 끝 표시는? ()
4. 다음과 같은 출력을 만드는 JSP 프로그램을 작성하시오. 단 마지막 두 줄의 계산 결과(35, 250)는 소스 코드에 직접 써넣지 말고 프로그램이 계산하여 출력하도록 하되, 표현식 `<%= %>`을 사용하시오.

공백을 5개 찍고 이 문장이 시작됩니다.

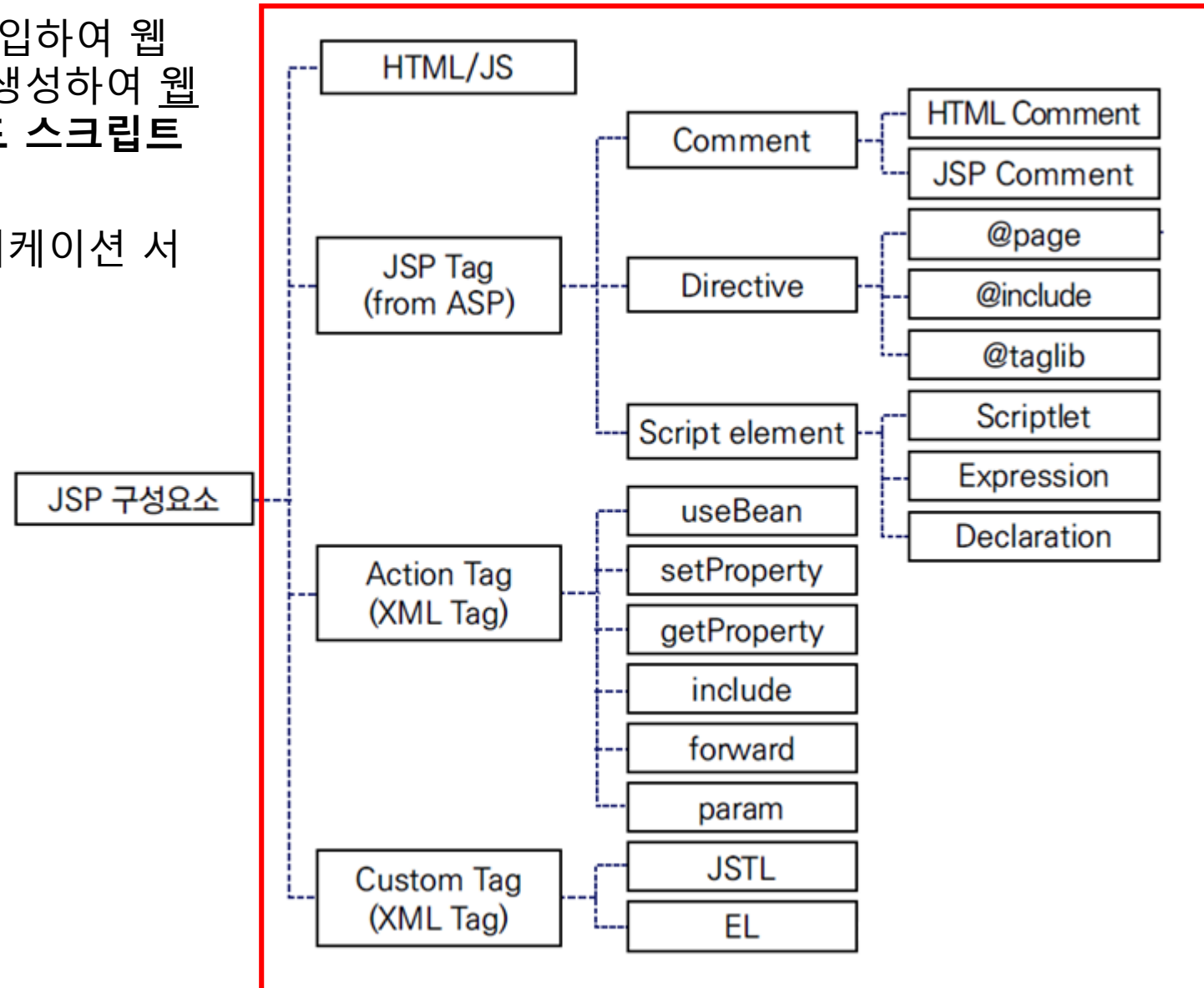
이 문장은 중간에 공백이 5개 있습니다.

$25 + 10 = 35$

$25 * 10 = 250$

JSP 구성 요소

- **JSP**는 HTML내에 **자바 코드**를 삽입하여 웹 서버에서 동적으로 웹 페이지를 생성하여 웹 브라우저에 돌려주는 **서버 사이드 스크립트 언어**
- Java EE 스펙 중 일부로 웹 애플리케이션 서버에서 동작



2.1 JSP 프로그램의 기본 형태

- JSP 프로그램 파일의 확장자는 “.jsp”
 - 파일의 내용은 기본적으로는 **HTML** 파일의 형태
 - 필요한 부분에 **JSP 코드들이 삽입**되는 형태
 - JSP 코드 시작을 알리는 표시로는 **<%** 끝을 알리는 표시로는 **%>**를 사용

2.1 JSP 프로그램의 기본 형태

[예제 2-1] JSP 프로그램의 기본 형태 (2-1.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"  
2:    pageEncoding="UTF-8"%>  
3: <%@ page import="java.time.*" %>  
4:  
5: <!DOCTYPE html>  
6: <html>  
7: <head>  
8:    <meta charset="UTF-8">  
9:    <title>테스트 페이지</title>  
10: </head>  
11: <body>  
12: <%  
13:    out.println("오늘 날짜 : " + LocalDate.now() + "<br>");  
14:    out.println("현재 시간 : " + LocalTime.now());  
15: %>  
16: </body>  
17: </html>
```

[실행 결과]

오늘 날짜 : 2018-08-21

현재 시간 : 11:41:19.174111200

2.1 JSP 프로그램의 기본 형태

1. page 지시자(Directive)

- <% 뒤에 @를 붙여서 <%@의 형태로 시작
 - 이렇게 <%@로 시작하는 코드를 지시자(directive)라고 함
 - 서블릿 컨테이너(톰캣)에게 무언가 **지시하거나 알려줄 것**이 있을 때 사용
- <%@ 뒤에 page라는 단어를 적은 것을 page 지시자라고 함
 - **JSP 페이지에 대한 정보를 톰캣에게 알려 줌**
 - 1: <%@ page **language**="java" **contentType**="text/html; charset=UTF-8"
 - 2: **pageEncoding**="UTF-8"%>

2.1 JSP 프로그램의 기본 형태

1. page 지시자(계속)

- 1: <%@ page language="java" contentType="text/html; charset=UTF-8"
- 2: pageEncoding="UTF-8"%>
- 3: <%@ page import="java.time.*" %>

속성	설명
info	<ul style="list-style-type: none">- 페이지를 설명해 주는 문자열(페이지 제목) <code><%@ page info="Business Management System" %></code>
language	<ul style="list-style-type: none">- JSP 페이지의 스크립트 요소에서 사용할 언어를 지정하는 속성 <code><%@ page language="java" %></code> 기본값 : java
contentType	<ul style="list-style-type: none">- HTTP 헤더에 그 내용이 들어감- 이 JSP 프로그램이 생성할 문서(실행된 결과)는 UTF-8로 인코딩 된 HTML <code><%@ page contentType="text/html; charset=utf-8" %></code> 기본값 : text/html
pageEncoding	<ul style="list-style-type: none">- JSP 페이지에 기록된 소스 코드 자체의 인코딩 방식 설정- 이 JSP 프로그램 파일은 UTF-8로 인코딩
import	<ul style="list-style-type: none">- 다른 패키지에 있는 클래스를 가져다 쓸 때 사용- 일반 자바 코드 <code>import "java.time.*";</code> 과 동일한 의미- import 속성은 page 지시자 중에 유일하게 중복 사용 가능

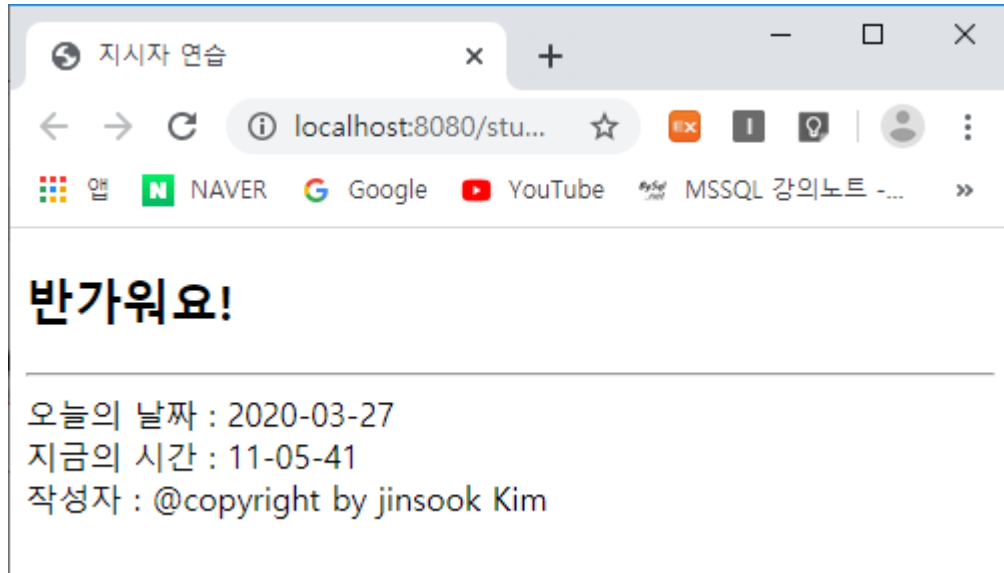
2.1 JSP 프로그램의 기본 형태

1. page 지시자(계속)

속성	설명
isThreadSafe	<ul style="list-style-type: none">• 하나의 JSP 페이지가 동시에 여러 브라우저의 요청을 처리할 수 있는지 여부를 설정• 멀티쓰레드 허용여부 설정
autoFlush	<ul style="list-style-type: none">• JSP 페이지 내용이 브라우저에 출력이 되기 전 버퍼에 저장되어 있는 내용 처리방법을 결정 하는 속성• 기본값은 true
buffer	<ul style="list-style-type: none">• JSP 페이지의 출력 크기를 킬로바이트(KB) 단위로 지정하는 속성• 기본값은 "8KB"(데이터 전송 단위)
session	<ul style="list-style-type: none">• HttpSession을 사용할지 여부를 지정하는 속성 (true가 기본값)
trimDirectiveWhitespaces	<ul style="list-style-type: none">• 결과 페이지(HTML)에서 빈 공간(JSP가 차지했던)을 없애는 속성

실습 - pageDirective

- 사용할 속성
 - info
 - import
 - trimDirectiveWhitespaces



2.1 JSP 프로그램의 기본 형태

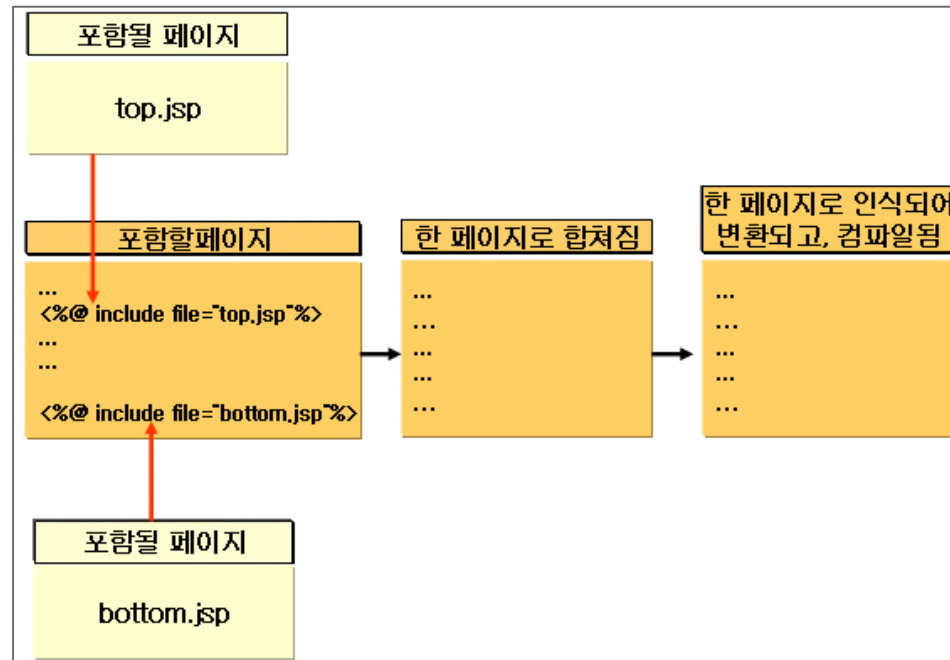
2. include 지시자

- 한 페이지 내에 다른 페이지를 삽입
 - 공통적으로 포함하는 내용을 가진 파일을 해당 JSP 페이지 내에 삽입하도록 하는 것
- include 디렉티브 사용법
 - `<%@ include file="포함될 파일의 URL"%>`
- include 디렉티브는 주로 **조각 코드**를 삽입할 때 사용

2.1 JSP 프로그램의 기본 형태

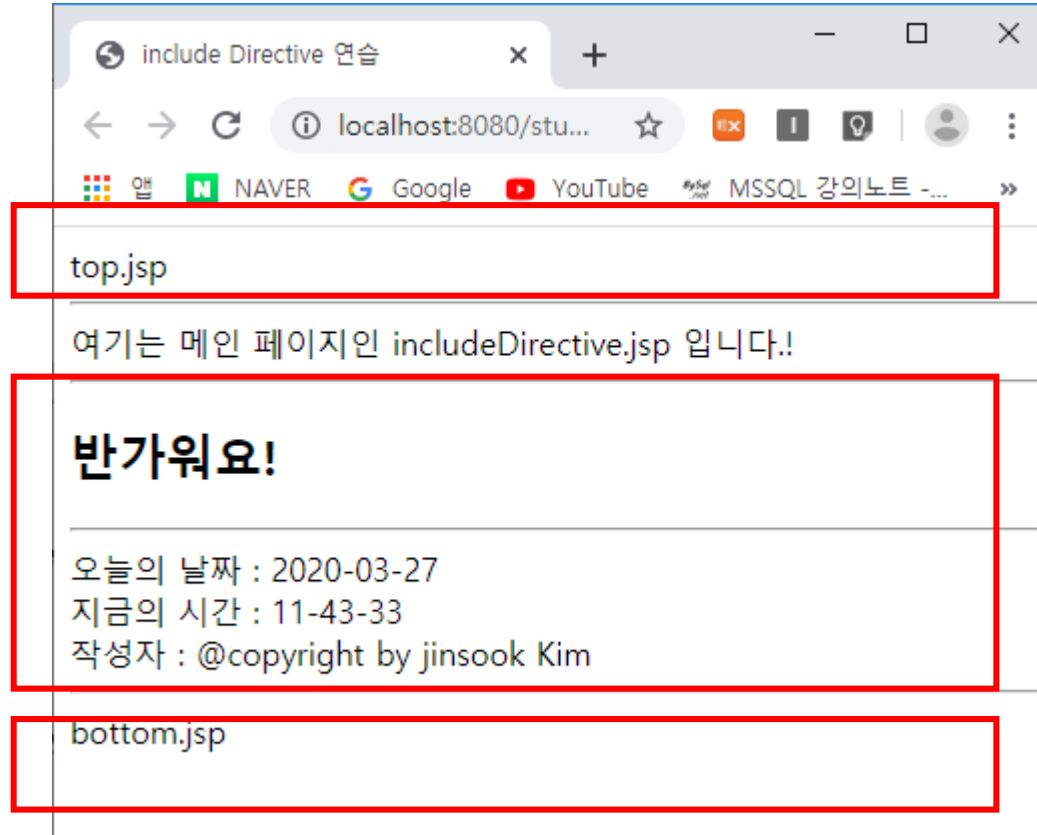
2. include 지시자(계속)

- include 디렉티브의 처리 과정은 **정적(static)**
 - include 지시자를 사용한 JSP 페이지가 컴파일 되는 과정에서 include 되는 JSP페이지의 소스 내용을 그대로 포함해서 컴파일됨
 - 복사 & 붙여넣기 방식으로 두 개의 파일이 하나의 파일로 구성된 후 같이 변환되고 컴파일



실습 - includeDirective

- includeDirective.jsp 에서 아래 3개의 파일을 include 하여 실행
 - top.jsp
 - pageDirective.jsp
 - bottom.jsp



2.1 JSP 프로그램의 기본 형태

3. 스크립틀릿(Scriptlet)

```
12: <%  
13:     out.println("오늘 날짜 : " + LocalDate.now() + "<br>");  
14:     out.println("현재 시간 : " + LocalTime.now());  
15: %>
```

- 자바 코드를 <% ... %> 사이에 적어놓은 것을 스크립틀릿이라고 부름
- 콘솔에 문자열을 출력할 때에는 System.out.print()나 System.out.println()를 쓰지만, 웹 브라우저에 출력을 하기 위해서는 out.print()나 out.println()을 사용

2.1 JSP 프로그램의 기본 형태

[예제 2-2] 2개의 스크립틀릿이 있는 프로그램 (2-2.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"
2:   pageEncoding="UTF-8"%>
3: <%@ page import="java.time.*" %>
4:
5: <!DOCTYPE html>
6: <html>
7: <head>
8:   <meta charset="UTF-8">
9:   <title>테스트 페이지</title>
10: </head>
11: <body>
12:
13: 오늘 날짜 :
14: <%
15:   out.println(LocalDate.now());
16: %>
17: <br>
18:
```

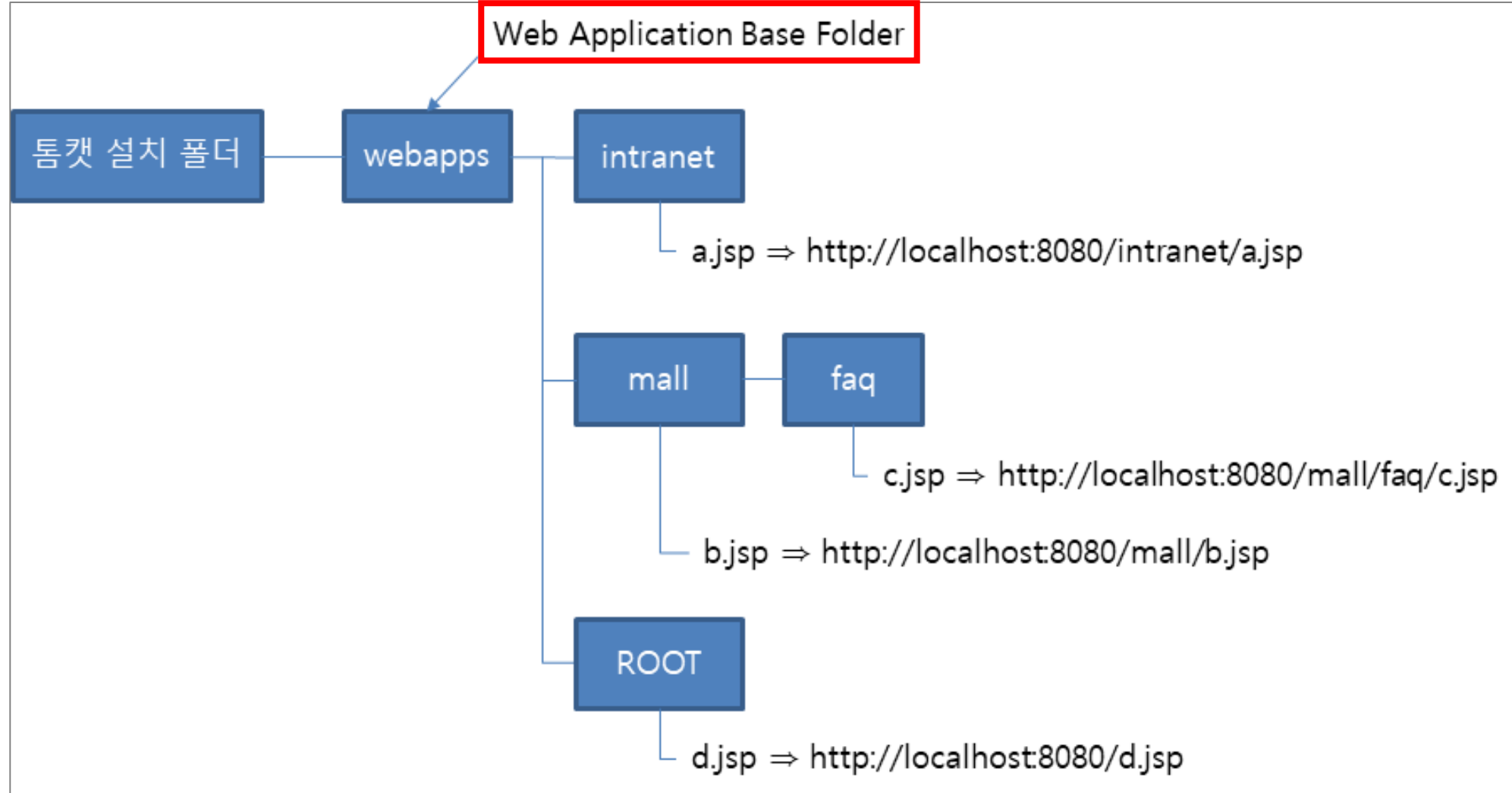
```
19: 현재 시간 :
20: <%
21:   out.println(LocalTime.now());
22: %>
23:
24: </body>
25: </html>
```

2.2 JSP 프로그램의 저장 위치와 실행 방법

- JSP는 HTML처럼 아이콘 더블 클릭/웹 브라우저 위에 드래깅해서 실행 안됨
 - **JSP 파일은 톰캣이 해석해야만 실행 결과를 얻을 수 있음**
 - 적절한 위치에 저장하고, 그에 맞는 URL를 브라우저 주소창에 입력해야만 정상적인 실행 가능
 - JSP 프로그램은 **톰캣의 애플리케이션 베이스(Application Base) 폴더의 하위 폴더에 저장**해야 함
- 애플리케이션 베이스(Application Base) 폴더
 - 톰캣이 설치된 폴더 아래에 있는 **webapps** 폴더 : C:\jsp\apache-tomcat-9.0.58\webapps
 - 하지만 이 폴더에 바로 jsp 파일을 넣으면 안 됨
 - 이 폴더는 톰캣이 관리하는 웹 애플리케이션을 모아두는 **루트 폴더**의 역할을 할 뿐
 - 웹 애플리케이션 : 하나의 웹 사이트를 구성하는 프로그램 덩어리
 - 예) 하나의 서버에서 인트라넷, 쇼핑몰, 두 개의 웹 애플리케이션을 관리

2.2 JSP 프로그램의 저장 위치와 실행 방법

- JSP 파일이 위치한 폴더와 이를 실행하기 위한 URL의 관계



2.3 주석(Comment)

- 자바의 여러 줄 주석(`/*...*/`)과 한 줄 주석(`//...`)은 스크립틀릿 영역에서 똑같이 사용
 - 이들 주석은 자바 영역(스크립틀릿)에서만 주석으로 인식
 - HTML 영역에서 이들 주석을 사용하면 일반 텍스트로 인식
- HTML 영역에서 주석 사용
 - `<!-- ... -->` 형태의 HTML 주석을 사용하거나,
 - `<%-- ... --%>` 형태의 JSP 주석을 사용

2.3 주석(Comment)

[예제 2-3] 주석을 추가한 프로그램 (2-3.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"
2:   pageEncoding="UTF-8"%>
3: <%@ page import="java.time.*" %>
4:
5: <!DOCTYPE html>
6: <html>
7: <head>
8:   <meta charset="UTF-8">
9:   <title>테스트 페이지</title>
10: </head>
11: <body>
12: <%
13:   /*
14:      여러 줄 주석 :
15:      Java 코드 시작
16:   */
17:   out.println("오늘 날짜 : " + LocalDate.now() + "<br>");
18:   out.println("현재 시간 : " + LocalTime.now());
19:   // 한 줄 주석 : Java 코드 끝
20: %>
```

2.3 주석(Comment)

```
21: <br> <br>
22: /* HTML 영역에서는
23:    JSP의 주석이 */
24: // 일반 텍스트로 인식됩니다.
25:
26: <!--
27:    HTML 영역에서는 HTML 주석을 사용하세요.
28:    이 주석은 "소스보기"를 하면 보입니다.
29: -->
30:
31: <%--
32:    HTML 영역에서 JSP 주석을 사용할 수도 있습니다.
33:    이 주석은 "소스보기"를 해도 보이지 않습니다.
34:    톱캣이 이 프로그램을 처리할 때 삭제하기 때문입니다.
35: --%>
36:
37: </body>
38: </html>
```

2.4 화면 출력

- JSP에서는 화면 출력을 위해 **out.print()** 또는 **out.println()**을 이용
 - System.out.print() 또는 System.out.println()과 같이 괄호 안에 출력하고 싶은 내용을 넣으면 됨
- 콘솔이 아니라 웹 브라우저 위에 출력되기 때문에 주의해야 할 점
 - 연속된 공백, 연속된 개행 문자, 또는 공백과 개행 문자들이 연속으로 섞여 있을 때는 **공백 하나로만 화면에 출력한다.**
 - 줄의 시작부터 나오는 공백은 출력하지 않고 무시한다.

22

[illegible][illegible]

연속된 공백은 공백 한 개로 출력
줄 시작부터 나오는 공백들은 무시됨
의도적인 공백

2.4 화면 출력

- 표현식(expression) : **<%= ... %>**
 - JSP를 이용해서 **값 하나만 출력**하고 싶은 경우에 유용하게 사용된다.
 - 변수값, 메소드 결과값 출력
 - 스크립트릿 내에서 표현식 사용 불가능
 - 스크립트릿 내에서 출력할 부분은 내장 객체인 out객체의 print()또는 println() 메소드를 사용
- 표현식내에서는 **세미콜론(;) 생략**
 - JSP페이지가 서블릿으로 변환될 때 표현식 부분은 out.write(); 메소드로 변환되어 자동적으로 세미콜론이 붙여지기 때문
- 표현식의 일반적인 문법
 - <%=문장%>
 - <%=name[i]%>

2.4 화면 출력

[예제 2-5] 표현식의 사용 (2-5.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"
2:   pageEncoding="UTF-8"%>
3: <%@ page import="java.time.*" %>
4:
5: <!DOCTYPE html>
6: <html>
7: <head>
8:   <meta charset="UTF-8">
9:   <title>테스트 페이지</title>
10: </head>
11: <body>
12:
13: 오늘 날짜 : <%= LocalDate.now() %> <br>
14: 현재 시간 : <%= LocalTime.now() %>
15:
16: </body>
17: </html>
```

답변할 수 있겠지요?

1. 스크립틀릿(JSP 프로그램 안에 있는 자바 코드)의 시작과 끝은 어떻게 표시하는지, 또 페이지 지시자의 시작과 끝은 어떻게 표시하는지 적어보시오.
2. 톰캣의 웹 애플리케이션 베이스 폴더(webapps)에 test라는 폴더를 만들고, 2-1.jsp이라는 이름의 프로그램 파일을 이 폴더에 넣었다고 할 때, 이 파일을 실행시키기 위해 웹 브라우저의 주소창에 써넣어야 하는 URL을 적어보시오.
3. 주석에 대한 다음 질문에 답해보시오.
 - ① 한 줄 주석의 시작 표시는? ()
 - ② 여러 줄 주석의 시작과 끝 표시는? ()
 - ③ JSP 주석의 시작과 끝 표시는? ()
 - ④ HTML 주석의 시작과 끝 표시는? ()
4. 다음과 같은 출력을 만드는 JSP 프로그램을 작성하시오. 단 마지막 두 줄의 계산 결과(35, 250)는 소스 코드에 직접 써넣지 말고 프로그램이 계산하여 출력하도록 하되, 표현식 `<%= %>`을 사용하시오.

공백을 5개 찍고 이 문장이 시작됩니다.

이 문장은 중간에 공백이 5개 있습니다.

$25 + 10 = 35$

$25 * 10 = 250$