

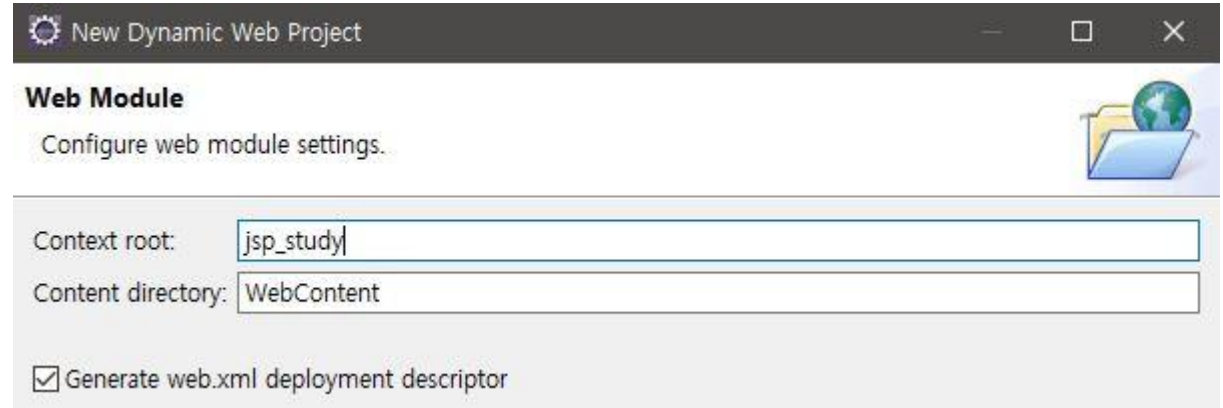
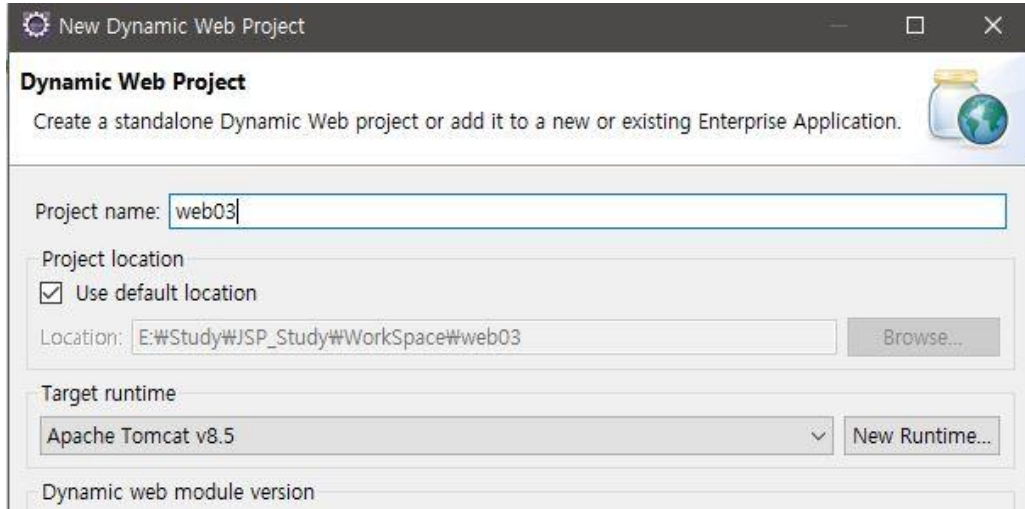
## 03강 JSP 기본 다루기

## JSP로 시작하는 웹 프로그래밍

- 인터넷은 사용자에게 정보를 제공하기 위해 HTML 태그를 사용하여 정보를 표시한다.
- JSP는 이런 HTML 태그를 사용할 수 있어서 웹 애플리케이션의 프레젠테이션(보여주기) 역할을 한다.
- 서버로부터 JSP를 요청하기 위해서는 해당 JSP가 속한 프로젝트가 서버에 컨텍스트에 등록되어 있어야 한다.
- 이클립스는 이러한 Context를 자동으로 등록해준다.

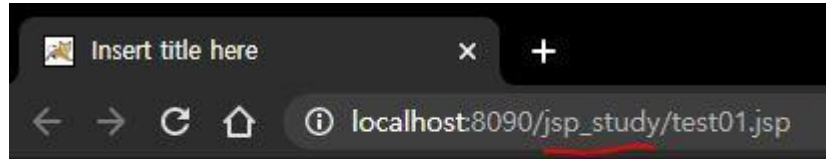
# JSP로 시작하는 웹 프로그래밍

- 그러나 처음 프로젝트를 만들 때 해당 Context를 변경할 수 있다.



# JSP로 시작하는 웹 프로그래밍

- 해당 프로젝트에 JSP 파일을 생성후 실행 해 본다.

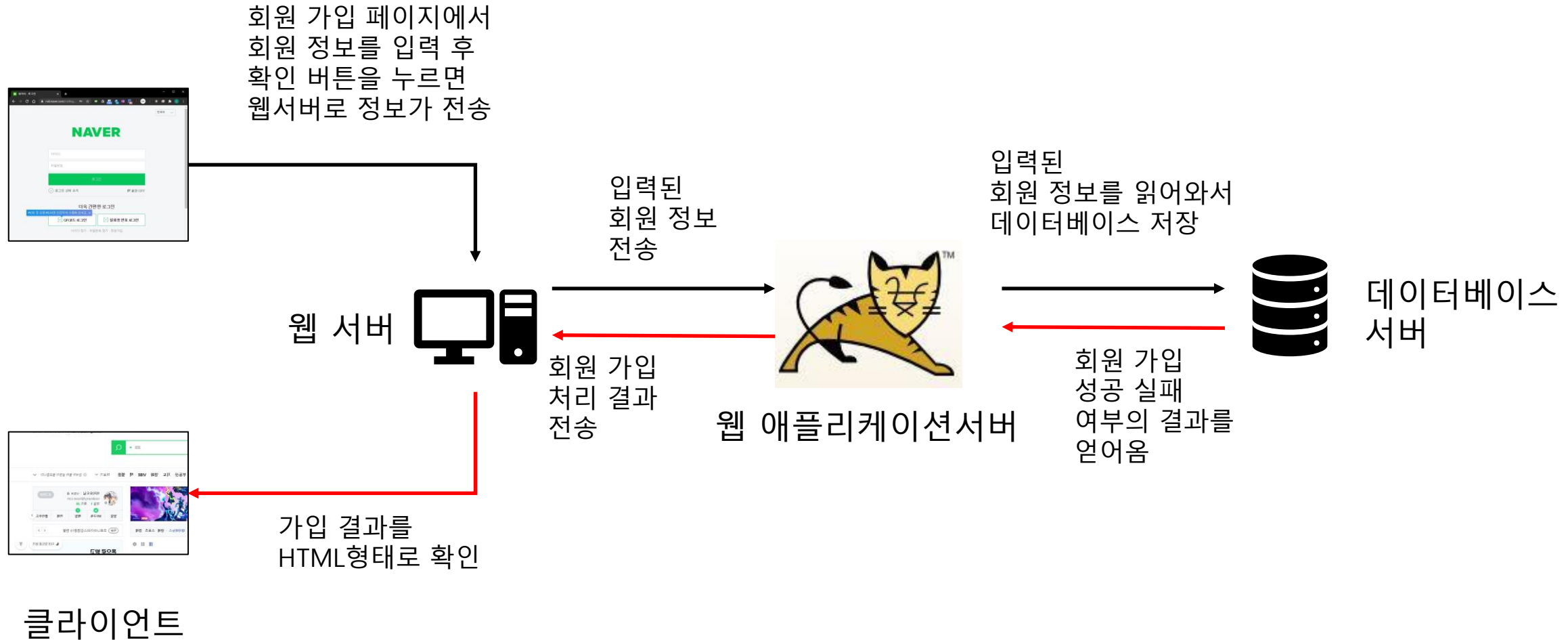


## Hello JSP

- Context Path가 프로젝트 이름과 다르게 등록된 이름으로 나오는 것을 볼 수 있다.
- 해당 이름은 서버의 server.xml에서 확인할 수 있다.

```
<Context docBase="web03" path="/jsp_study"
```

# JSP로 시작하는 웹 프로그래밍



## JSP로 시작하는 웹 프로그래밍

- JSP은 기존 HTML과 다른 방식으로 동작한다.
- HTML은 확장자가 html이고 클라이언트의 웹브라우저 내의 번역기가 해석하는 반면 JSP는 확장자가 jsp이고 WAS가 해석해서 그 결과를 HTML로 번역한 후 클라이언트에 전송한다.
- JSP는 HTML내부에 <% %>태그 내부에 작성한다,. =>예제 1
- <%%>내부의 코드를 서블릿내의 자바 코드로 변환한다.

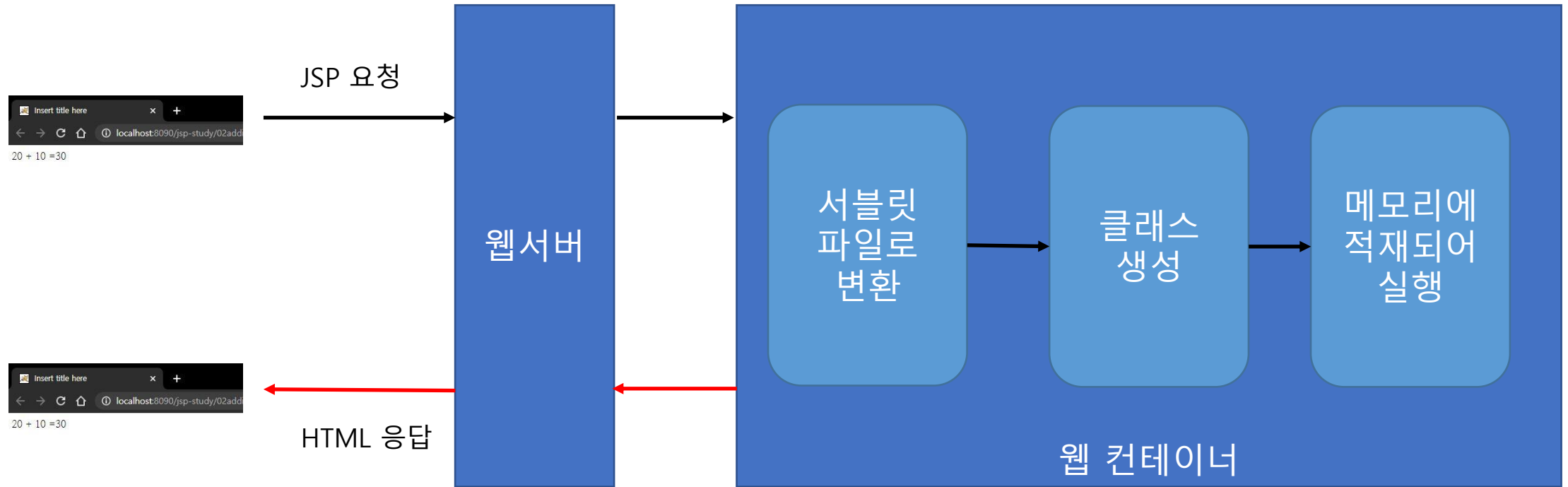
# JSP로 시작하는 웹 프로그래밍

- 서블릿과 JSP의 차이
- 자바 기반의 웹 애플리케이션을 구동하기 위해서는 서블릿이 필요하다.
- 그러나 서블릿은 자바 코드 내에 HTML문서 구조가 담기는 형태라서 자바를 모르는 웹 개발자가 보기에 불편한다.
- 그래서 HTML코드 기반의 JSP가 탄생했다.
- 웹 개발자가 보기에 서블릿보다 JSP가 쉽고 간편하다.
- 다만 서블릿 컨테이너는 JSP를 해석하지 못하므로 JSP를 서블릿으로 변환해주어야 한다.

JSP가 서블릿으로 변환된 코드를 알고 싶다면 다음 위치를 찾아본다.

work\Catalina\localhost\jsp-study\org\apache\jsp

# JSP로 시작하는 웹 프로그래밍

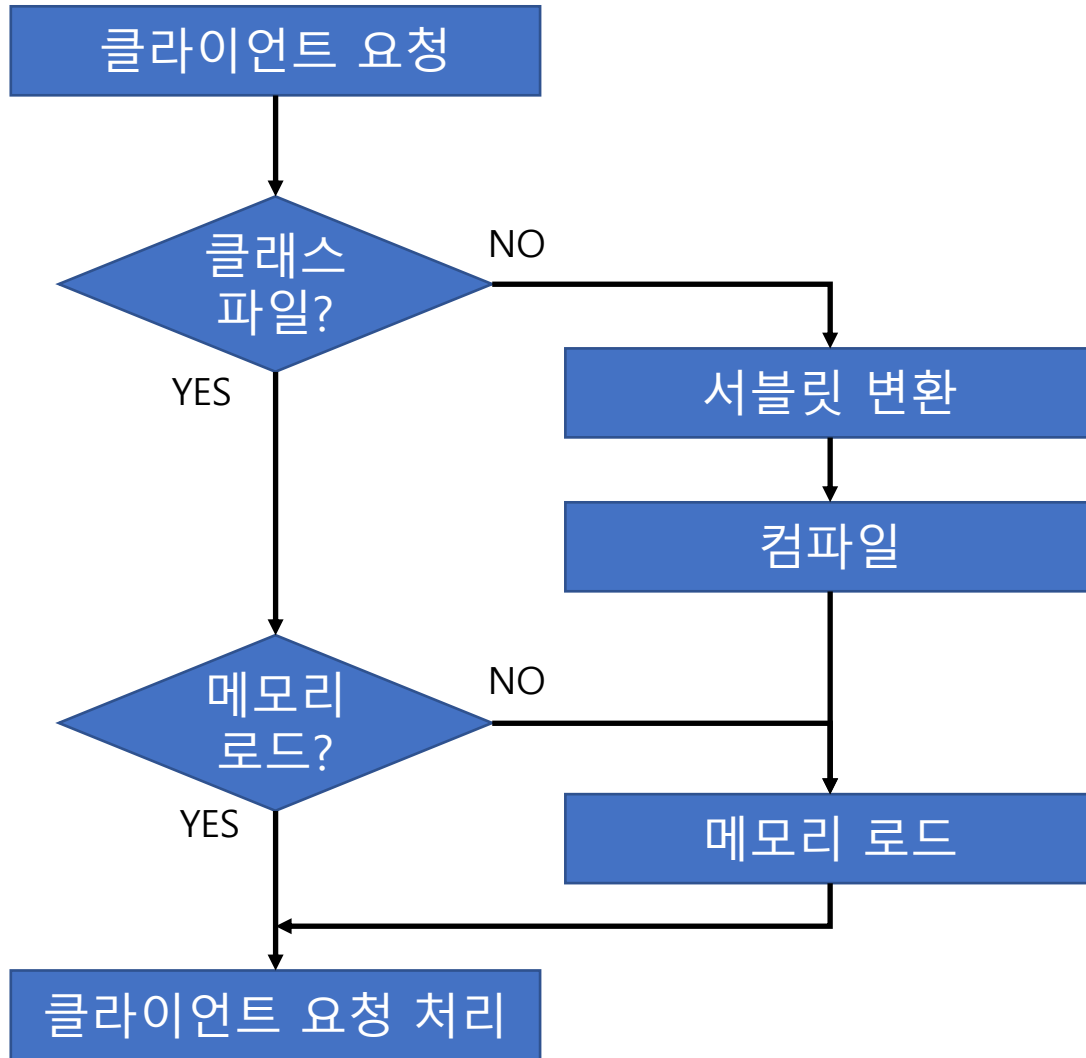


클라이언트

서버



# JSP로 시작하는 웹 프로그래밍



# JSP 기본 태그

- 기본 HTML이나 자바스크립트에 의해 작성된 페이지는 정적 페이지이다.
  - (예외 존재)
- 동적 페이지를 구성하기 위해 JSP 태그를 사용해야 한다.

종류	사용용도	형식
스크립트릿	자바 코드를 기술함	<% %>
선언	변수와 메소드를 선언함	<%! %>
표현식	계산식이나 함수를 호출한 결과를 문자열 형태로 출력함	<%= %>
주석	JSP 페이지에 설명을 넣음	<%-- --%>
지시자	JSP 페이지의 속성을 지정함	<%@ %>

- JSP문서는 JSP 확장자가 붙는다.
- 이는 서버에서 컴파일이 되며 HTML태그도 함께 사용된다.

## JSP 기본 태그 - 스크립트릿

- JSP에서 자바코드를 기술하기 위해서는 스크립트릿 태그를 사용해야 한다.
- 기본 태그의 형태는 <% %>이다.

```
<%  
int num1 = 20;  
int num2 = 10;  
int add = num1 + num2;  
out.print( num1 + " + " + num2 + " =" + add);  
%>
```

## JSP 기본 태그 - 선언문

- JSP 페이지 전체에 사용될 변수나 메소드의 선언은 선언문내부에 작성한다.
- 선언문의 형태는 <%!     %>
- <% %>에 선언하면 \_jspService()메소드 내부에 들어가므로 메소드 내부에 메소드나 static 변수를 선언할 수 없다.

```
<%!String str = "안녕하세요!";  
    int a = 5, b = -5;  
  
    public int abs(int n) {  
        if (n < 0) {  
            n = -n;  
        }  
        return n;  
    }%>
```

```
<%  
    out.print(str + "<br>");  
    out.print(a + "의 절대값 : " + abs(a) + "<br>");  
    out.print(b + "의 절대값 : " + abs(b) + "<br>");  
%>
```

## JSP 기본 태그 - 선언문

- 선언문 내부에 선언된 변수는 인스턴스 변수이고
- 스크립트릿에 선언된 변수는 로컬 변수이다.

## JSP 기본 태그 - 표현식

- HTML형태로 결과를 출력하기 위해서는 JSP 내장 객체중 출력을 담당하는 out의 print메소드를 사용한다.

```
<%  
    out.print(a);  
%>
```

- 그러나 매번 이런 식으로 출력하기에 코드가 불편하므로 단순 출력은 표현식을 사용해서 출력한다.
- 표현식 <%= %> 형태로 사용

```
<%=변수 %>  
<%=수식 %>  
<%=메소드 호출 %>
```

- 다만 반드시 ;를 표시하면 안된다.

## JSP 기본 태그 - 주석문

- JSP에도 주석을 남길수 있다.
- JSP 주석문은 `<%-- --%>` 형태로 기술한다.
- HTML의 주석문과 JSP주석문은 기본적으로 실행에 아무런 영향을 주지 않는다..
- 다만 차이가 있다면 HTML주석은 내용 그대로 클라이언트에 보내지만 JSP의 주석은 클라이언트로 보내지 않는다.
- 클라이언트에 공개해선 안되는 정보는 JSP 주석을 사용한다.
- 스크립트릿 내부에 자바 주석을 사용할 수도 있다
  - `//` 한줄 주석
  - `/*` 블록 주석 `*/`

## JSP 기본 태그 - 지시자

- 지시자는 JSP 페이지 전체에 적용될 설정을 적용시키기 위한 태그이다.
- 사용법은 <%@ 지시자 속성 %> 형태로 사용한다.
- 지시자의 속성은 page, include, taglib가 있다

종류	사용 용도
page	해당 JSP 페이지 전반적으로 환경을 설정할 내용을 지정한다.
include	현재 페이지에 다른 파일의 내용을 삽입할 때 사용한다.
taglib	태그 라이브러리에서 태그를 꺼내와 사용할 수 있는 기능을 제공한다.



## JSP 기본 태그 – 지시자 - page

- page 지시자는 JSP 페이지에 여러 정보를 나타내기 위해서 사용되는 JSP문법
- page 지시자의 기본 형식

<%@ page 속성=값 ...%>

- 속성에는 여러가지 내용이 올 수 있다

- 속성 종류

- language
- extends
- import
- session
- buffer
- autoFlush
- isThreadSafe
- info
- errorPage
- isErrorPage
- contentType

## JSP 기본 태그 – 지시자 - page

- language => JSP에서 사용되는 언어를 결정    무조건 JAVA만 기술할 수 있다.(생략가능)
- extend   => JSP 내부에서 특정 클래스로부터 상속을 받으려면 사용해야 한다.
- import   => 자바에서 사용하는 import와 같은 역할이다.
- session   => 세션을 사용할지 말지 결정하는 속성 기본값은 true
- buffer   => 대용량 데이터를 전송하기 위해서 출력버퍼에 데이터가 쌓일 때 까지 기다렸다가 클라이언트에 제공하는 기능을 한다. 기본값은 버퍼의 크기를 8KB로 설정하는데 버퍼 용량을 크게하면 오히려 버퍼가 찰 때까지 기다리므로 오히려 전송이 느려질 수 있다. 버퍼를 사용하지 않는 경우 none 설정을 한다.
- autoFlush   => 버퍼를 비운다. autoFlush가 기본적으로 true가 되어있는데 버퍼가 차면 자동으로 비운다. 만약 false면 버퍼가 가득찬 경우 에러를 발생시킨다. buffer=none과 autoflush를 false를 적용하면 반드시 에러가 발생하므로    동시 적용하면 안된다..
- isThreadSafe   => JSP는 기본적으로 자바를 기반으로 하므로 멀티스레드를 지원한다. 다만 멀티스레드 상황에서 여러 사용자가 동시 요청시 동시에 스레드가 생성되어 문제가 발생할 수 있다. 이럴때를 대비해서 isThreadSafe속성을 true로 설정하면 동기화를 통해서 각페이지의 스레드가 안전하게 동작할 수 있다,

## JSP 기본 태그 – 지시자 - page

- info => JSP 페이지에 대한 간략한 설명을 기록하는 용도로 사용한다.
- errorpage => JSP 페이지에서 에러가 발생했을때 보여줄 에러 페이지를 결정한다.
- isErrorPage => 현재 페이지가 에러페이지인지 아닌지 알려주는 용도로 사용되는 속성, 값이 기본값으로 false이며 true일 경우 exception 객체로 예외의 원인을 알려주는 용도로 사용된다.

⇒예제 2

- contentType=>JSP 페이지의 MIME 타입을 결정한다. 브라우저에게 데이터 정보를 헤더에 포함해서 알려주는 용도로 사용한다. 보통 HTML문서를 보내주므로 text/html값을 지정한다. 추가로 인코딩방식으로 지정할 수 있는데 ;다음으로 charset=인코딩 방식, 주로 UTF-8을 사용한다.

## JSP 기본 태그 – 지시자 - include

- include지시자는 현재 페이지에 다른 HTML문서나 JSP 페이지의 내용을 삽입할 때 사용
- 유일한 속성은 file인데 URL 주소를 속성값으로 가진다.
- 모든 페이지에 공통으로 보여 주어야 할 부분이 있다면 적용한다.

⇒예제3

- 단 하나의 페이지로 합쳐서 컴파일이 진행되므로 각 페이지에 생성된 변수중복이 발생하면 안된다.
- taglib는 사용자 정의 태그를 만들어 사용할 때 사용하는데 주로 사용자 정의 태그를 모아둔 jar 파일을 적용할 때 사용한다.