

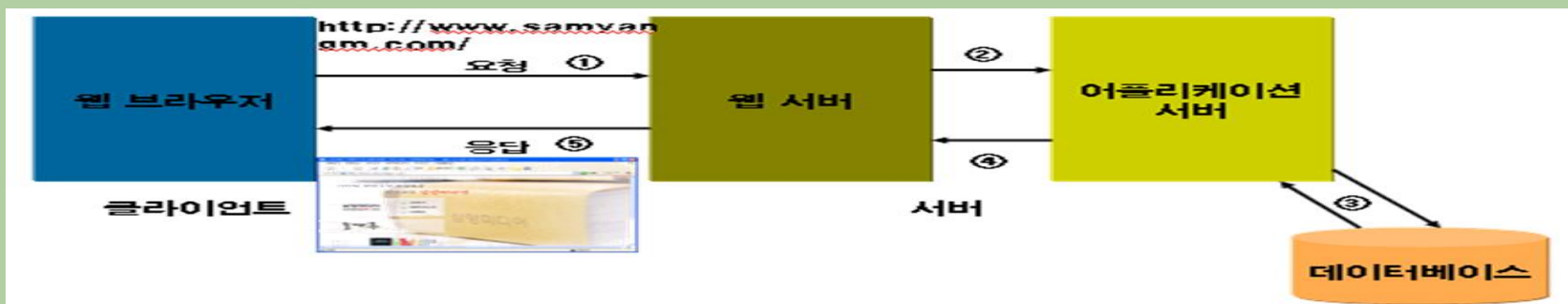


# **JSP(JavaServer Page)**

# JSP 필수 구성요소

## 웹 어플리케이션 구축을 위해 필요한 구성들

- ▷ 웹 서버 : 웹 브라우저의 요청을 받아 알맞은 결과를 웹 브라우저에 전송
- ▷ 어플리케이션 서버 : 게시글 목록, 로그인 처리와 같은 기능을 실행, 그 결과를 응답으로 웹 서버로 전달
- ▷ 데이터베이스 : 웹 어플리케이션이 필요로 하는 데이터를 저장한다. 예를 들어, 회원정보, 게시판 글, 데이터 등을 저장한다
- ▷ 웹 브라우저 : 웹 서버에 서비스 실행을 요청하며, 웹 서버의 처리 결과를 사용자에게 보여준다



# JSP 동작과정

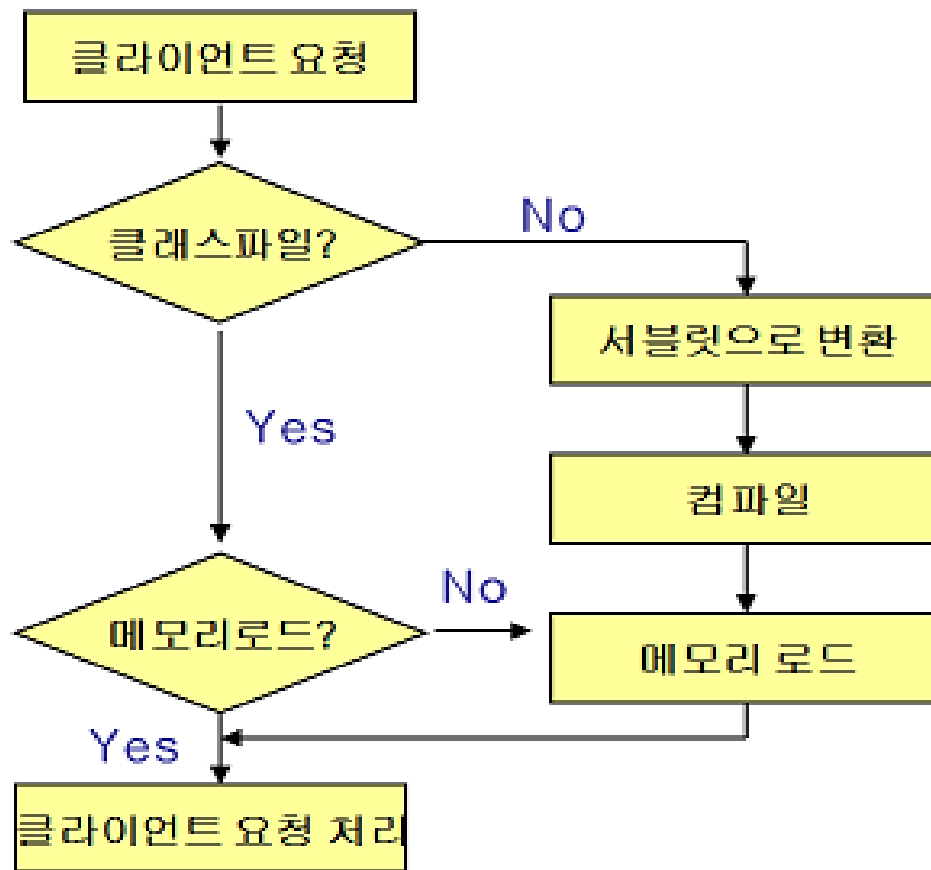


# JSP 동작과정

- 웹 브라우저에서 jsp페이지를 웹 서버로 요청하게 되면, 웹 서버는 JSP에 대한 요청을 웹 컨테이너로 넘김
- 이런 요청을 받은 웹 컨테이너는 해당 JSP 페이지를 찾아서 서블릿(자바 파일)으로 변환을 시킨 다음 서블릿(자바 파일)을 컴파일
- 컴파일된 서블릿(자바 파일)은 최종적으로 웹 브라우저 응답되어져 사용자는 응답 결과를 보게됨
- 이러한 과정은 해당 jsp페이지가 최초로 요청되었을 때 단 한번만 실행. 이후 같은 페이지에 대한 요청이 있으면 변환된 서블릿 파일로 서비스를 처리

# JSP 동작과정

- JSP → 서블릿 코드로 변환 (work 폴더에 저장)

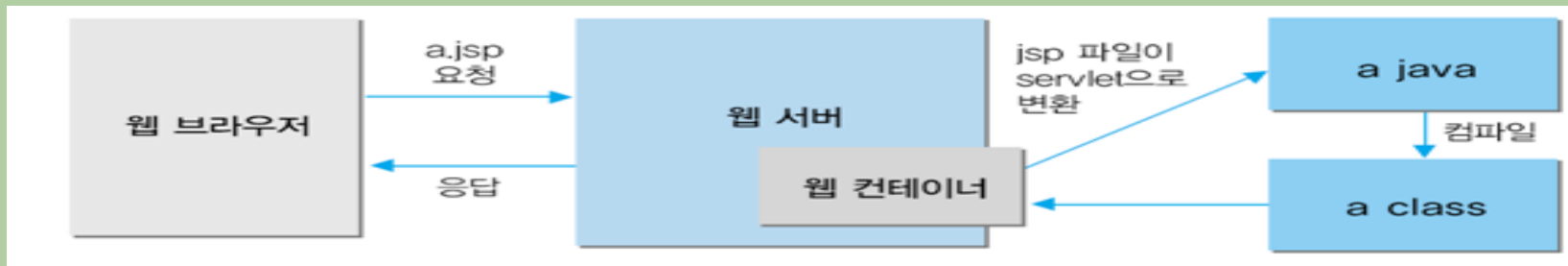


1. 서블릿 코드 생성 단계
2. 컴파일 단계
3. 요청 처리 단계

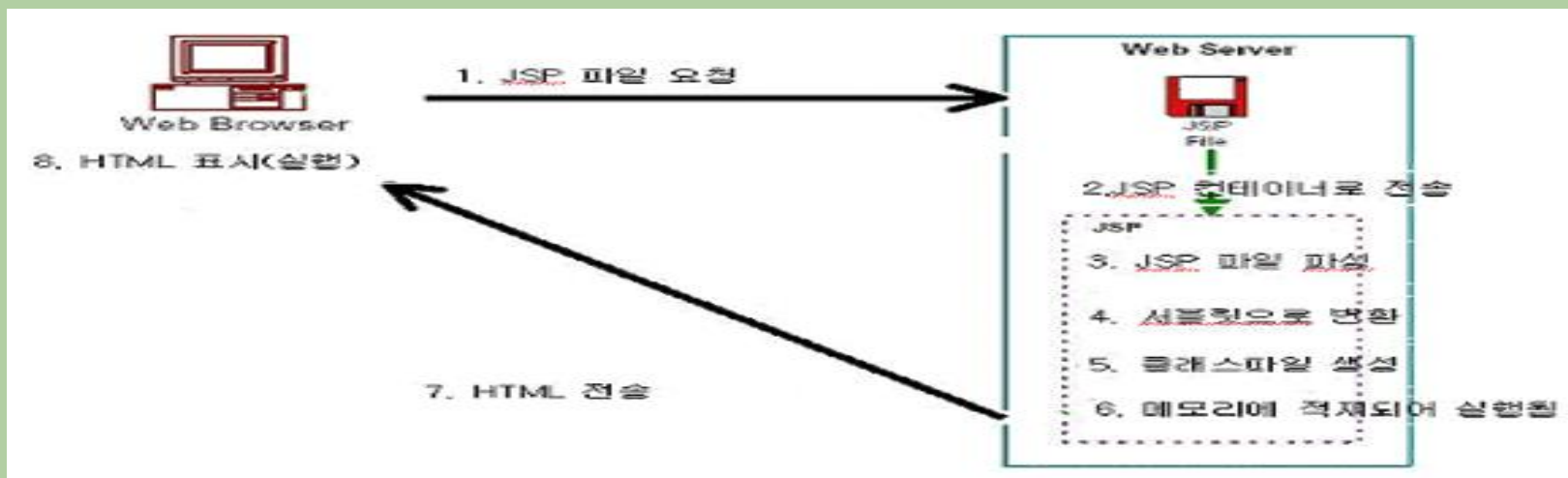


# JSP 동작과정

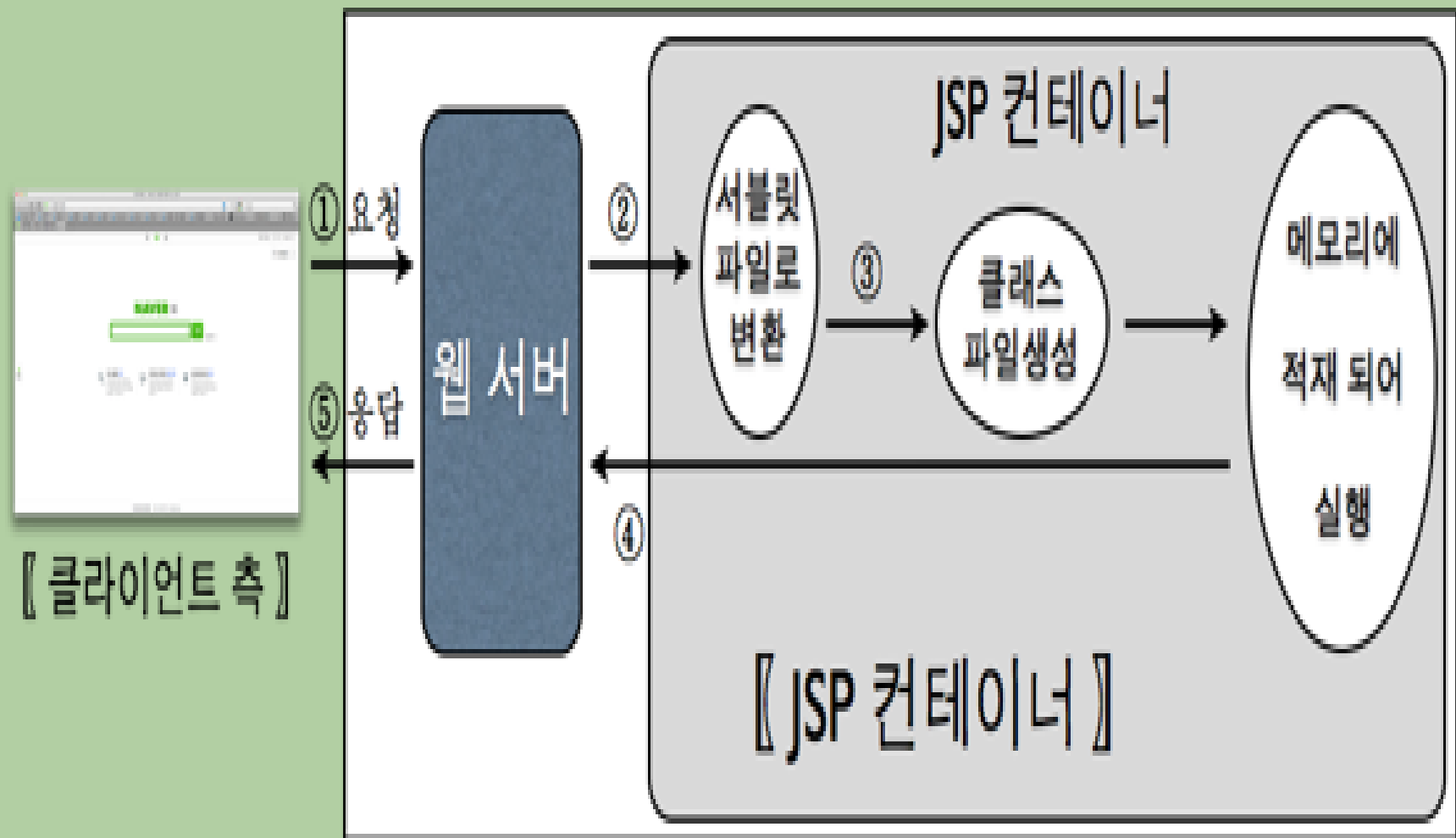
jsp페이지가 서블릿으로 변화하는 과정을 단계별로 표시



jsp페이지가 서블릿으로 변화하는 과정을 단계별로 표시



# JSP 동작과정



# JSP 와 서블릿

## ▷ 서블릿 :

- 자바 언어를 개발한 썬마이크로에서 웹 개발을 위해 만든 표준이 서블릿(Servlet)이다
- 서블릿을 만들기 위해 자바 코드를 작성, 코드를 컴파일 해서 클래스 파일을 만들게 됨
- 서블릿은 실행 코드 방식에 속하여 데이터를 조금 바꾸고 싶어도 코드를 수정하고 컴파일하고 클래스를 알맞은 곳에 복사해주는 작업 때문에 JSP를 만들게 됨

## ▷ JSP :

- 서블릿 표준을 기반으로 만들어졌다. JSP 파일이 번역되면 최종 결과물로 서블릿 생성
- JSP(JavaServer Pages)는 스크립트 언어로서 다음과 같은 특징을 가지고 있다
  - 자바 언어를 기반으로 하는 스크립트 언어로서 자바가 제공하는 기능을 그대로 사용
  - HTTP와 같은 프로토콜에 따라 클라이언트의 요청을 처리하고 응답
  - HTML, XML 등 클라이언트가 요청한 문서를 생성하는 데 주로 사용
  - 서블릿 / EJB 등의 엔터프라이즈 기술들과 잘 융합
  - 표현언어, 표현식, 스크립트릿 등 다양한 스크립트 요소와 액션 태그 등을 제공



# JSP 사용이유

1. 자바 언어를 기반으로 하고 있기 때문에 플랫폼에 상관없이 사용
2. 자바 언어에 대한 깊은 이해가 없어도 괜찮다
3. 대규모 어플리케이션을 구현할 때 사용되는 스프링과 같은 프레임워크와 완벽하게 연동 되며, 금융권에서 많이 사용되는 다른 엔터프라이즈 기술과도 완벽하게 연동됨

# JSP 시작

웹 브라우저

1. 웹 브라우저는 웹 서버에 어떤 기능을 원하는지 요청

요청

request

2. 웹 서버는 웹 어플리케이션을 실행해 웹 브라우저가 요청한 기능을 수행한 후 결과를 웹 브라우저에 응답한다.

웹 서버

처리

웹 어플리케이션

HTML

응답

response

웹 브라우저

3. 웹 브라우저는 웹 서버로부터의 응답 결과를 출력

# JSP 시작

## JSP 페이지의 구성요소

- ❶ 디렉티브(Directive)
- ❷ 스크립트(Script) :  
스크립트릿(Scriptlet), 표현식(Expression), 선언부(Declaration)
- ❸ 표현 언어(Expression Language)
- ❹ 기본 객체(Implicit Object)
- ❺ 정적인 데이터
- ❻ 표준 액션 태그(Action Tag)
- ❼ 커스텀 태그(Custom Tag)와 표준 태그 라이브러리(JSTL)

# JSP 지시자(PAGE, TAGLIB, INCLUDE)

page	JSP 페이지에 대한 정보를 지정. 생성하는 문서의 타입, 출력 버퍼의 크기, 여러 페이지 등을 지정한다.
taglib	JSP 페이지에서 사용할 태그 라이브러리를 지정한다.
include	JSP 페이지의 특정 영역에 다른 문서를 포함시킨다.

# JSP 사용법

## 스크립트 요소

표현식 (Expression)	값을 출력한다. <b>&lt;%= %&gt;</b>
스크립트릿 (Scriptlet)	자바 코드를 작성하고 실행한다. <b>&lt;% %&gt;</b>
선언부 (Declaration)	자바의 메서드를 만들어 선언한다. <b>&lt;%! %&gt;</b>

# JSP 사용법

## 스크립트 요소

종 류	사 용 용 도	형 식
지시자(Directive)	JSP 페이지의 속성을 지정함	<%@ %>
주석(Comment)	JSP 페이지 설명을 넣음	<% - ->
선언(declaration)	변수와 메소드를 선언함	<%! %>
표현식(expression)	계산식이나 함수를 호출한 결과를 문자열 형태로 출력함	<%= %>
스크립트릿(scriptlet)	자바 코드를 기술함	<% %>
액션 태그(Action Tag)	자바 빈이나 애플릿 서버 모듈과 연결함	<jsp:action> </jsp:action>



# JSP 사용법

## 기본 객체

기본 객체	실제 타입	설 명
<b>request</b>	javax.servlet.http.HttpServletRequest 또는 javax.servlet.ServletRequest	클라이언트의 요청 정보를 저장한다.
<b>response</b>	javax.servlet.http.HttpServletResponse 또는 javax.servlet.ServletResponse	응답 정보를 저장한다.
<b>pageContext</b>	javax.servlet.jsp.PageContext	JSP 페이지에 대한 정보를 저장한다.
<b>session</b>	javax.servlet.http.HttpSession	HTTP 세션 정보를 저장한다.
<b>application</b>	javax.servlet.ServletContext	웹 어플리케이션에 대한 정보를 저장한다.
<b>out</b>	javax.servlet.jsp.JspWriter	JSP 페이지가 생성하는 결과를 출력할 때 사용되는 출력 스트림이다.
<b>config</b>	javax.servlet.ServletConfig	JSP 페이지에 대한 설정 정보를 저장한다.
<b>page</b>	java.lang.Object	JSP 페이지를 구현한 자바 클래스 인스턴스이다.
<b>exception</b>	java.lang.Throwable	예외 객체. 에러 페이지에서만 사용된다.

# JSP 사용법

## 표현 언어

EL은 다양한 위치에 있는 데이터에 접근하기 위한 언어이다. 문법체계가 직관적으로 아주 쉽다. JSP에서는 모든 변수의 생성과 선언을 반드시 표시를 해주어야 하지만 EL은 그과정 없이 바로 접근 가능하다.

- 1) JSTL (jsp standard tag library) 1.0 규약에 소개 된 내용
- 2) JSP 2.0 에 새롭게 추가된 스크립트 언어
- 3) 기존의 Script Tag의 표현식Tag의 업그레이드된 출력태그

### ◆사용하는 이유

표현언어를 사용하면 표현식을 사용하는 것보다 간단한 방법으로 값을 출력할 수 있음

### ◆ 제공 기능

- 1) jsp 의 네가지 기본객체 (page, request, session, application) 가 제공하는 scope의 속성사용
- 2) 집합 객체(배열, 컬렉션)에 대한 접근 방법 제공
- 3) 수치연산 , 관계 연산, 논리 연산자 제공
- 4) java class method 호출 기능 제공
- 5) 표현 언어만의 기본 객체 제공

### ◆기본문법

- 1) \${표현식}
- 2) 표현식에는 변수명or속성명.메소드 구조로 이루어짐
- 3) 표현식에는 정수형, 실수형, 문자열형, 논리형, null 이 올 수 있음

# JSP 사용법

액션 태그	<p>JSP 페이지 내에서 특별한 기능을 제공한다. 액션 태그는 &lt;jsp:액션태그이름&gt;의 형태를 띠며 여러가지 종류가 있다.</p> <p>ex) &lt;jsp:include&gt;의 경우 특정 페이지의 실행 결과를 현재 위치에 포함시킨다.</p>
커스텀 태그	<p>액션 태그처럼 태그 형태로 기능을 제공하지만 사용자가 직접 개발해 주어야 한다는 차이점이 있다.</p>
태그 라이브러리	<p>커스텀 태그 중에서 자주 사용하는 것들을 별도로 표준화한 것이다. 이것이 JSTL 이다.</p> <p>조건문과 반복문을 커스텀 태그를 이용하여 구현할 수 있도록 해준다. 스크립트 코드보다 이해하기 쉽기 때문에 어느 정도 논리적인 연산을 처리할 수 있다.</p>

# JSP 지시자 (PAGE)

속성	값	기본값	용례
info	텍스트	문자열 없음	info="Registration form"
language	스크립트 언어이름	"java"	language="java"
contentType	MIME 타입 문자집합		contentType="text/html; charset=ISO-8859-1"
extends	클래스 이름	없음	extends="com.taglib.wd jsp.MyJspPage"
import	클래스/패키지 이름	없음	import="java.net.URL" import="java.util.*,java.text.*"
session	불린 플래그	"true"	session="true"
buffer	버퍼크기, 아니면 none	"8kb"	buffer="12kb" buffer="none"
autoFlush	불린 플래그	"true"	autoFlush="false"
isTreadSafe	불린 플래그	"true"	isThreadSafe="true"
errorPage	로컬 URL	없음	errorPage="result/failed.jsp"
isError	불린 플래그	"false"	isErrorPage="false"

# JSP 내장 객체(기본객체)

내장 객체	리턴 타입	설명
request	javax.servlet.http.HttpServletRequest 또는 javax.servlet.ServletRequest	클라이언트의 요청 정보를 저장하고 있는 객체이다.
response	javax.servlet.http.HttpServletResponse 또는 javax.servlet.ServletResponse	클라이언트의 요청에 대한 응답 정보를 저장하고 있는 객체이다.
out	javax.servlet.jsp.JspWriter	JSP페이지 출력할 내용을 가지고 있는 출력 스트림 객체이다.
session	javax.servlet.http.HttpSession	세션 정보를 저장하고 있는 객체이다.
application	javax.servlet.ServletContext	웹 어플리케이션 context의 정보를 저장하고 있는 객체이다.
pageContext	javax.servlet.jsp.PageContext	JSP페이지 대한 정보를 저장하고 있는 객체이다.
pageContext	java.lang.Object	JSP페이지를 구현한 자바 클래스 객체이다.
config	javax.servlet.ServletConfig	JSP페이지 대한 설정정보를 저장하고 있는 객체이다.
exception	java.lang.Throwable	JSP페이지에서 예외가 발생시에만 사용되는 객체이다.

# JSP 내장 객체(REQUEST)

- JSP 에서 가장 많이 사용되는 기본 객체로서 웹 브라우저의 요청과 관련이 있다.
- 클라이언트가 요청한 정보를 제공하는 것이 request 기본 객체이다.

다음과 같은 기능을 가지고 있다.

- 클라이언트(웹 브라우저)와 관련된 정보 읽기 기능
- 서버와 관련된 정보 읽기 기능
- 클라이언트가 전송한 요청 파라미터/헤더/쿠키 읽기 기능
- 속성 처리 기능



# JSP 내장 객체(REQUEST)

## 클라이언트 정보 및 서버 정보 읽기

- 클라이언트가 전송한 정보 및 서버정보를 구할 수 있는 메서드를 제공한다.

<code>getRemoteAddr()</code>	클라이언트의 IP주소
<code>getContentTypeLength()</code>	클라이언트 요청정보의 길이. 알수 없는 경우 -1 리턴
<code>getCharacterEncoding()</code>	클라이언트가 요청정보를 전송할 때 사용한 캐릭터의 인코딩
<code>getContentType()</code>	클라이언트가 사용한 contentType
<code>getProtocol()</code>	클라이언트가 요청한 프로토콜
<code>getMethod()</code>	웹 브라우저가 정보를 전송할 때 사용한 방식
<code>getRequestURI()</code>	웹 브라우저가 요청한 URL 경로
<code>getContextPath()</code>	JSP 페이지가 속한 웹 어플리케이션의 컨텍스트 경로
<code>getServerName()</code>	서버 이름
<code>getServerPort()</code>	서버가 실행 중인 포트 번호

# JSP 내장 객체(REQUEST)

## HTML 폼과 요청 파라미터의 처리

- 웹 브라우저 폼에 입력한 값을 처리하는 것은 가장 많이 사용하는 기능 중 하나이다. request 기본 객체는 전송한 파라미터를 읽어올 수 있는 메서드를 제공한다.

getParameter(String name)	이름이 name인 파라미터의 값. 존재하지 않을 경우 null
getParameterValues(String name)	이름이 name인 모든 파라미터의 값을 배열로 구한다. 존재하지 않을 경우 null
getParameterNames()	웹 브라우저가 전송한 파라미터의 이름
getParameterMap()	웹 브라우저가 전송한 파라미터의 맵

# GET , POST

## get 방식과 post 방식

- get 방식으로 전송할 경우 URL 뒤에 쿼리문자열이 생성된다. 즉, get 방식은 굳이 폼을 사용하지 않더라도 파라미터를 전송할 수 있다. post 방식은 데이터 영역을 이용하여 전송한다.
- get 방식은 웹 브라우저, 웹 서버 또는 웹 컨테이너에 따라 전송할 수 있는 파라미터 값의 길이에 제한이 있을 수 있다. post 방식은 데이터 영역을 이용하여 전송하기 때문에 파라미터 길이에 제한이 없다.

# JSP 내장 객체(RESPONSE)

- request 기본 객체와 정반대의 기능을 수행한다.  
즉, 웹 브라우저에 보내는 응답 정보를 담는다.

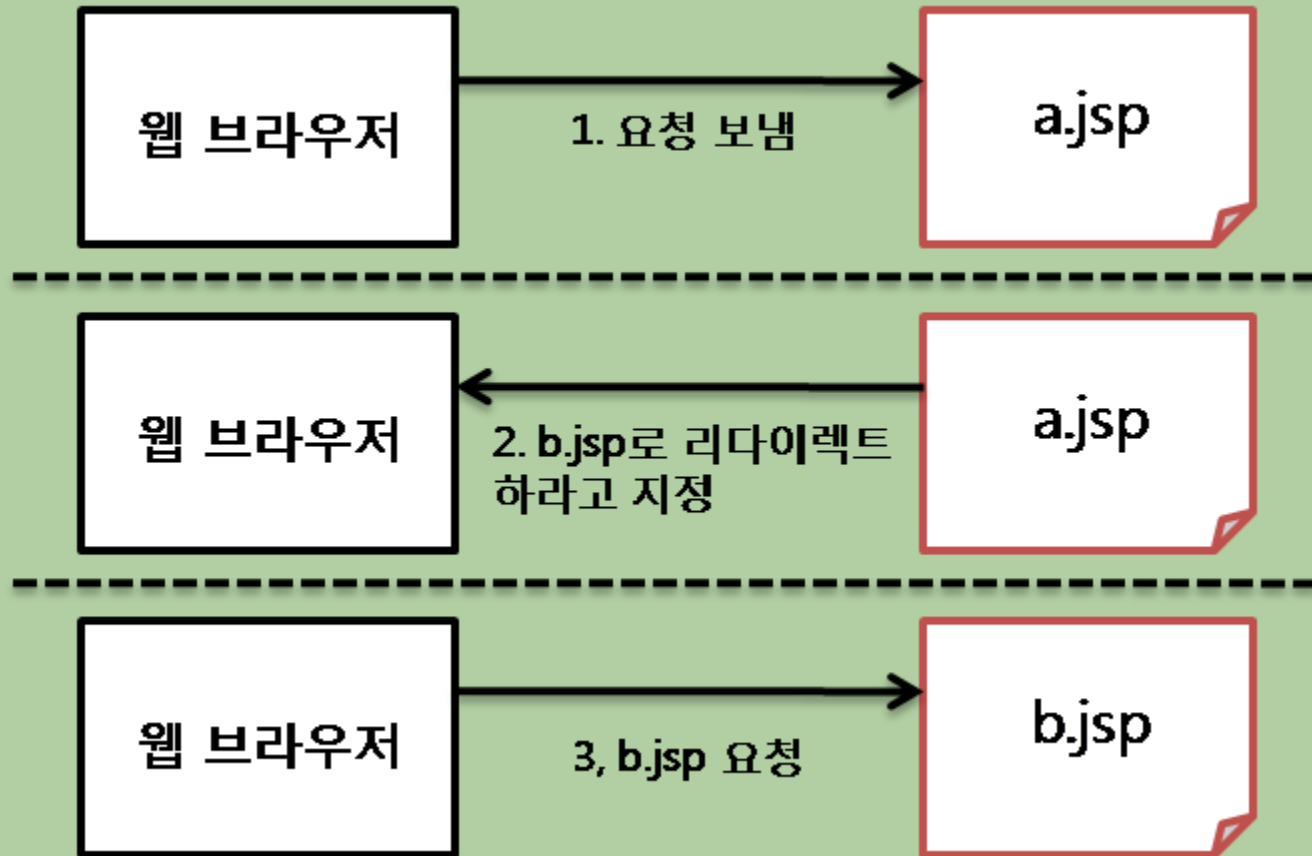
- 리다이렉트를 이용한 페이지 이동

- 리다이렉트는 response 기본 객체에서 많이 사용되는 기능 중 하나이다.

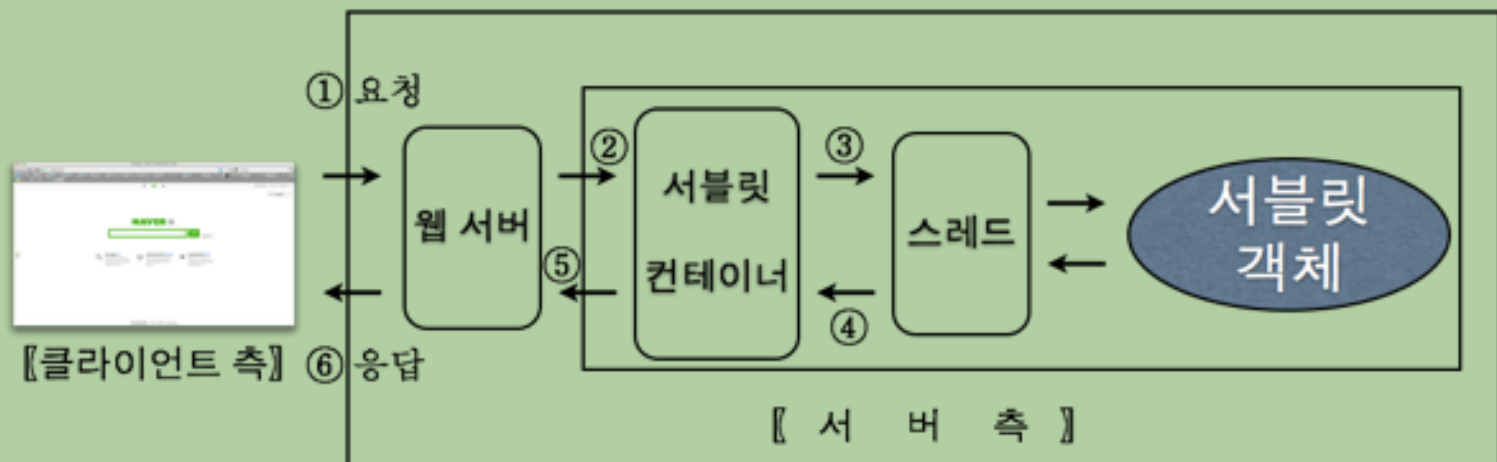
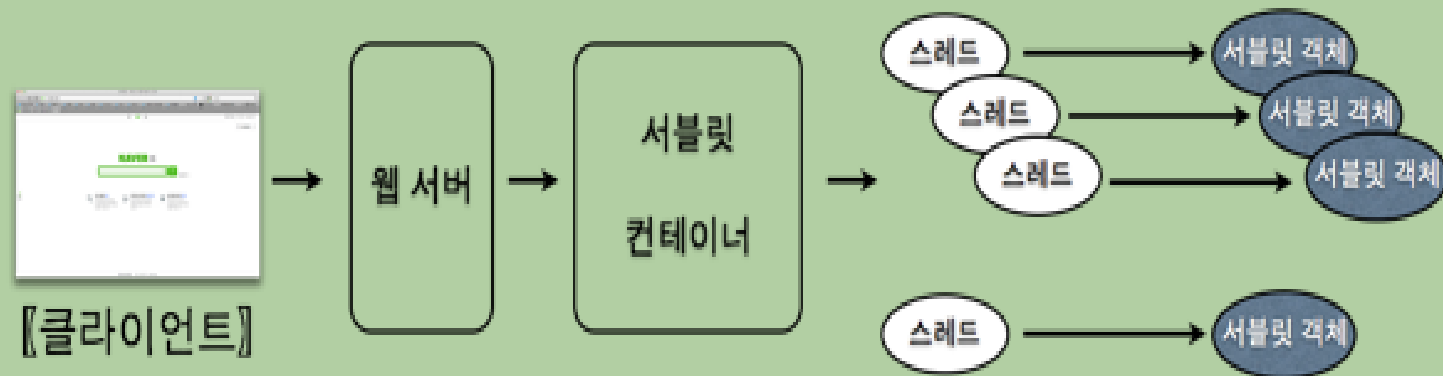
리다이렉트 기능은 웹 서버가 다른 페이지로 이동하라고 지시하는 것을 의미 한다.

특정 페이지를 실행 한 후 지정한 페이지로 이동하길 원할 때 사용한다.

# JSP 내장 객체(RESPONSE)



# JSP 실제 동작





# 기타 (GET,POST)

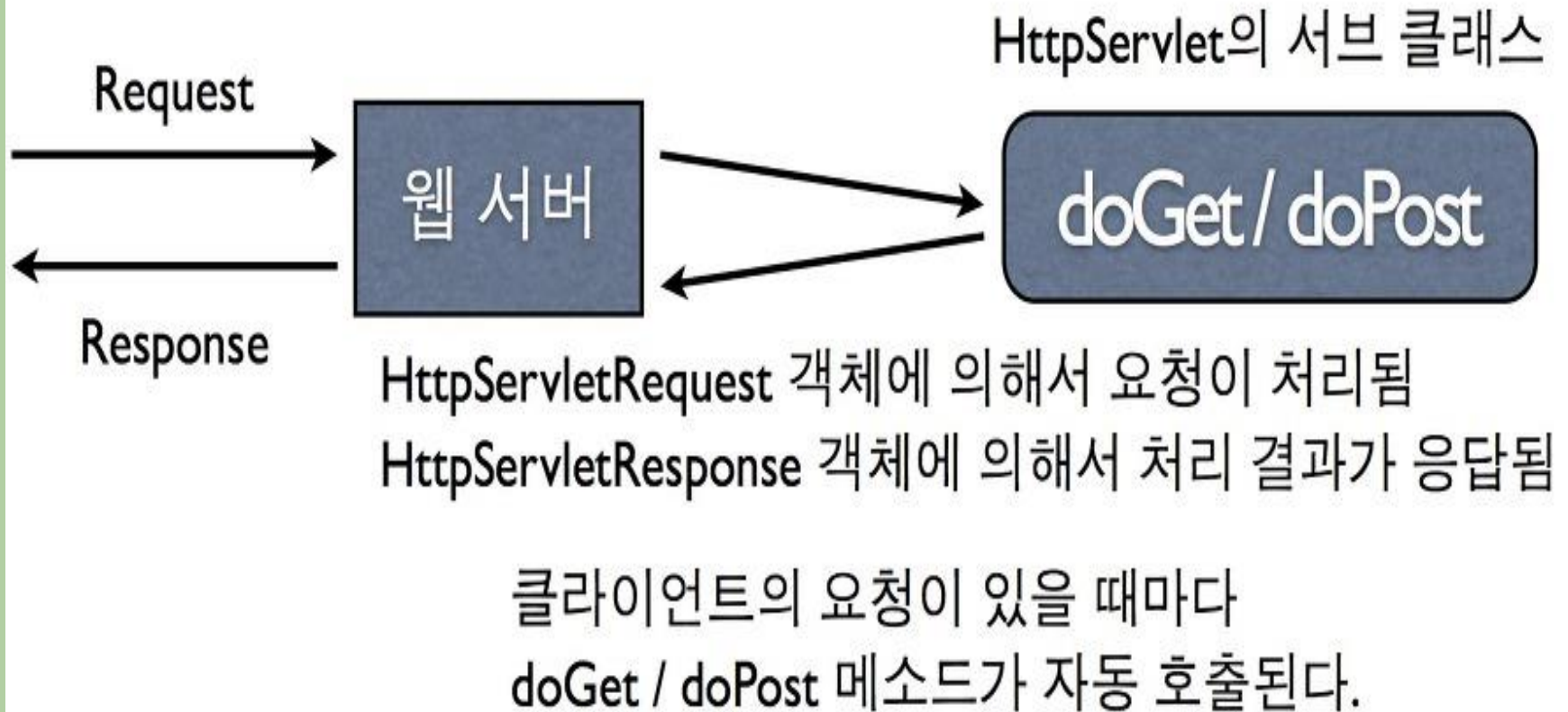
## GET 방식

- 주소 창을 타고 넘어가기 때문에 보안이 되지 않는다.
- 255자 이하의 소용량 데이터를 전송한다.
- GET 방식으로 요청하면 doGet 메소드가 호출된다.

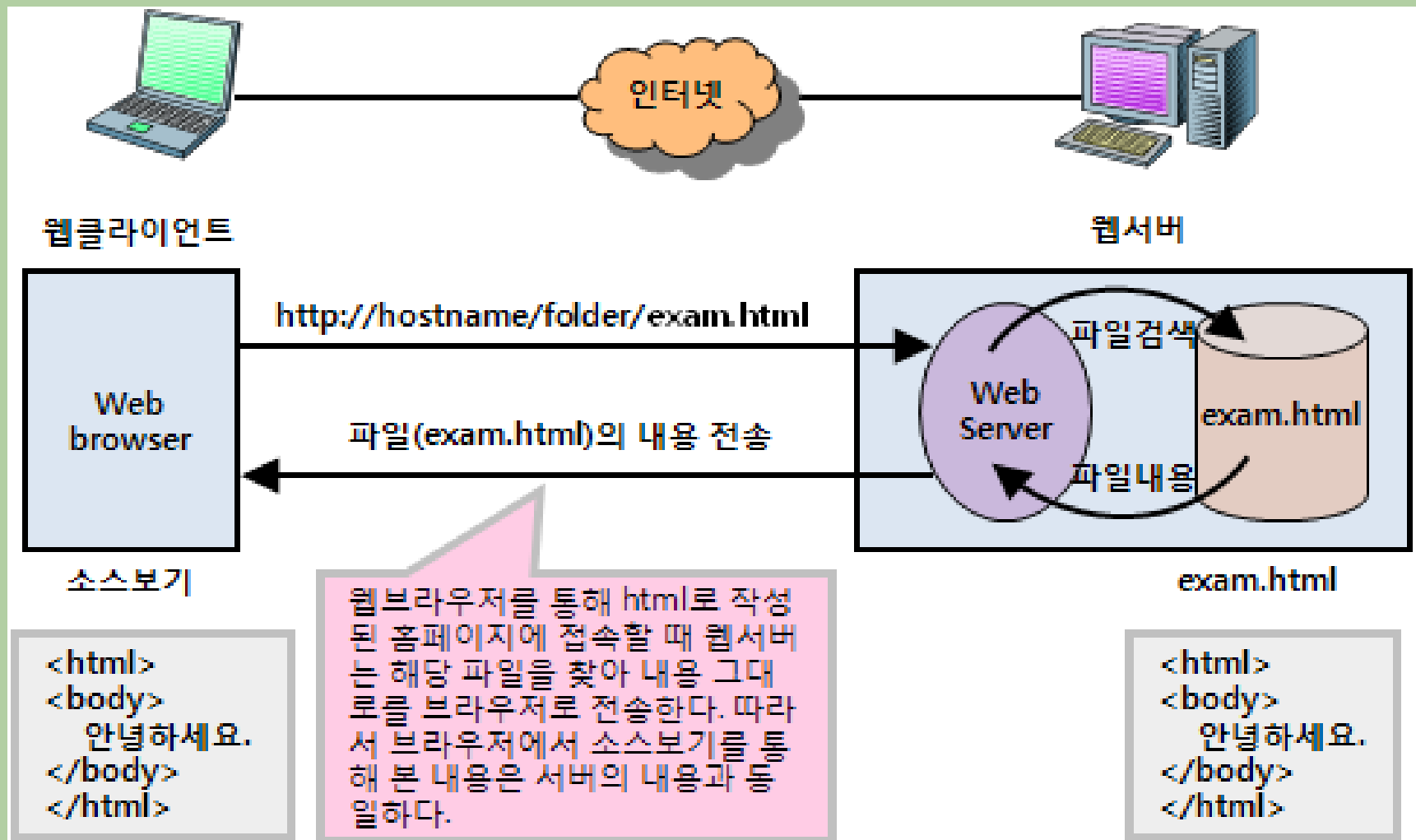
## POST 방식

- html header를 타고 넘어가기 때문에 보안이 된다.
- 255자 이상의 대용량 데이터를 전송한다.
- POST 방식으로 doPost 데이터를 전송한다.

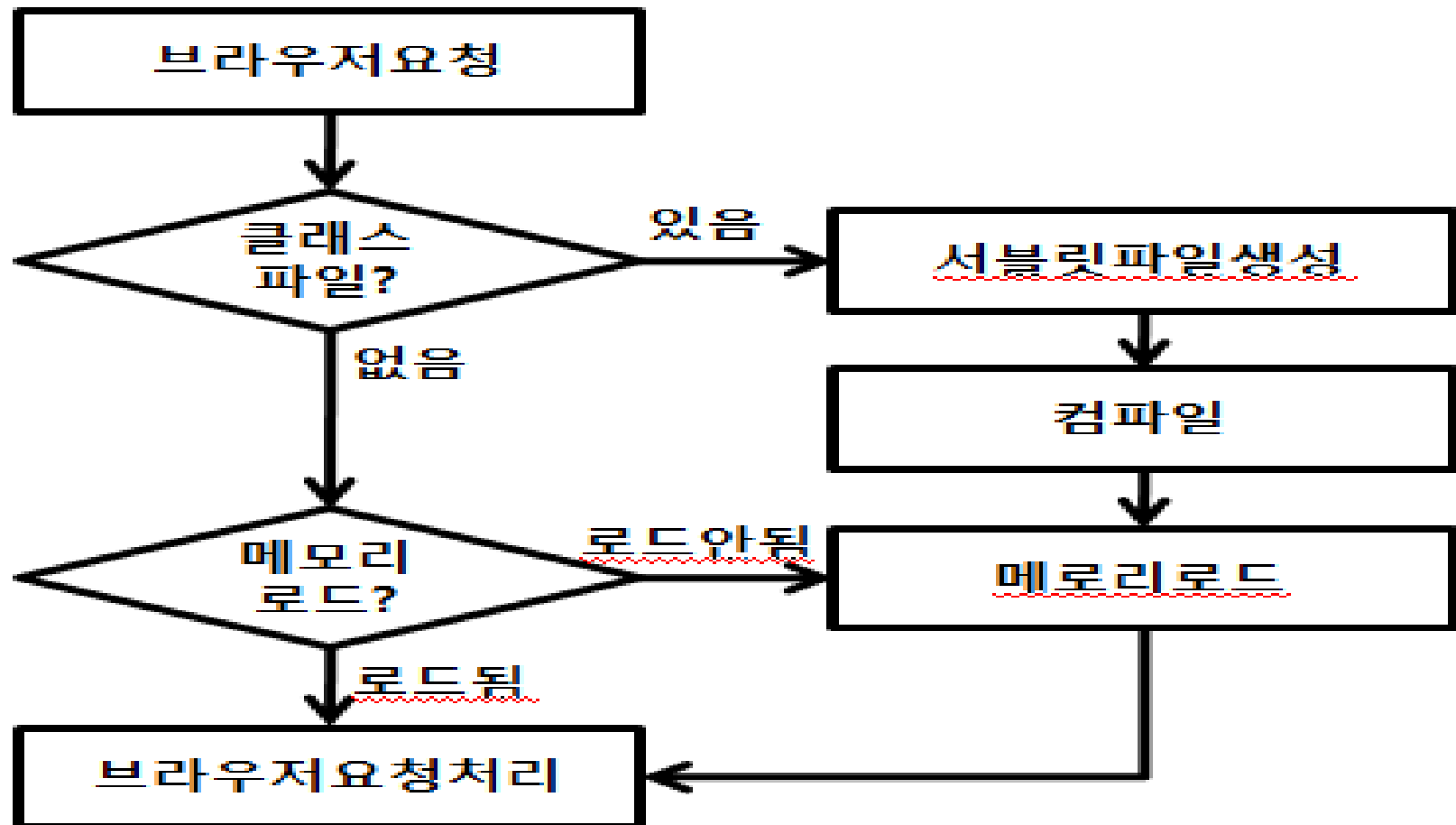
# 기타 (GET,POST)



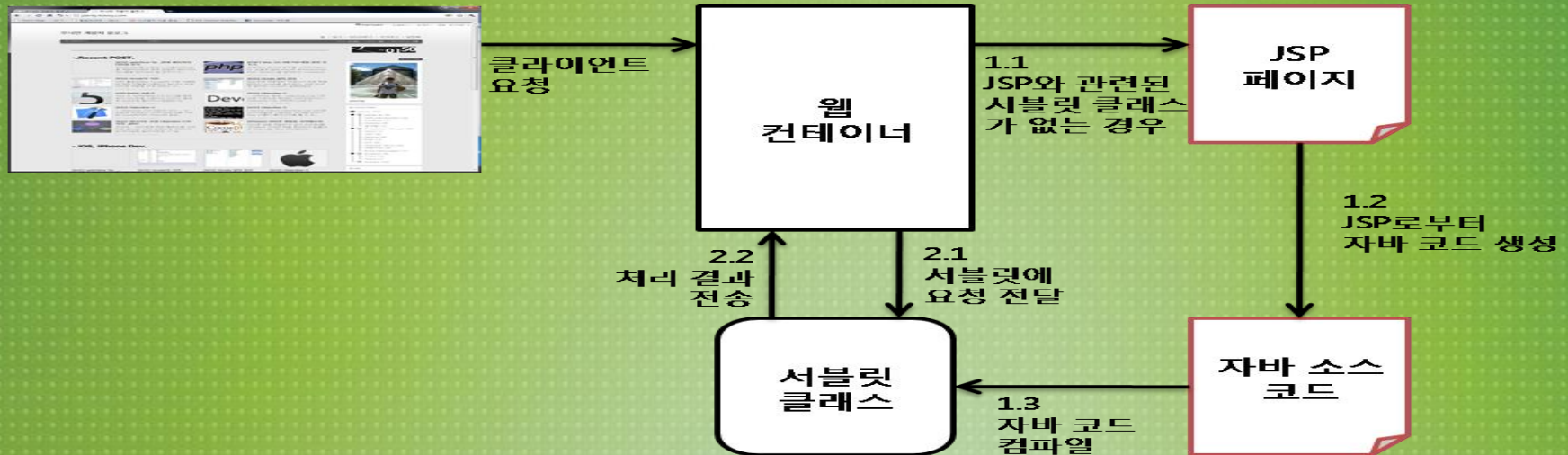
# 기타



# 기타



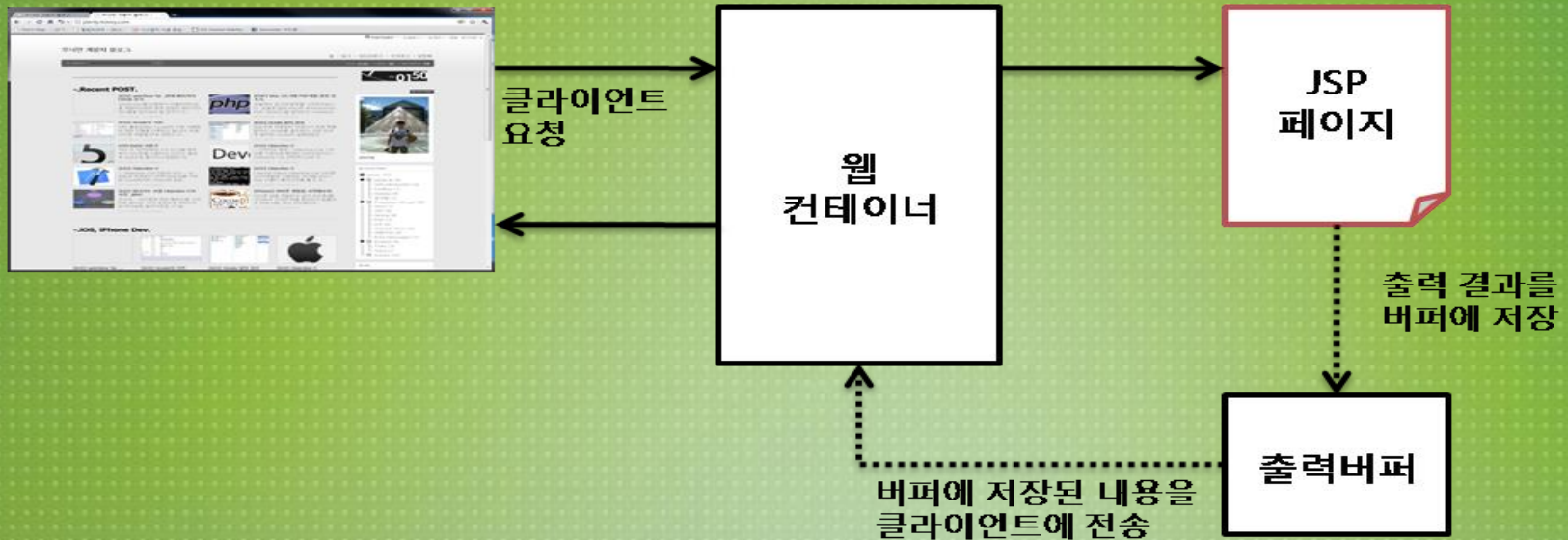
# JSP의 처리 과정



- JSP에 해당하는 서블릿이 존재하지 않을 경우 JSP페이지를 컴파일 하여 서블릿을 생성한 후, 생성 된 서블릿을 사용하여 클라이언트의 요청을 처리.
- JSP에 해당하는 서블릿이 존재하는 경우, 곧바로 서블릿을 사용하여 클라이언트의 요청을 처리한다.



# JSP 출력버퍼와 응답



- ❶ JSP 페이지는 생성된 결과를 곧바로 웹 브라우저에 전송하지 않고, 출력 버퍼라고 불리는 곳에 임시로 출력 결과를 저장했다가 한 번에 웹 브라우저 전송한다.
- ❷ JSP는 기본적으로 출력 내용을 버퍼에 저장한 후 나중에 전송한다.