



Chapter01

웹 프로그래밍의 이해



Contents

학습목표

- ❖ 웹 프로그래밍 전반적인 환경을 이해한다.
- ❖ JSP의 기술 및 동작방식에 대해 이해한다.
- ❖ JSP 공부 방법과 알아야 하는 기술 및 수준을 알아본다.

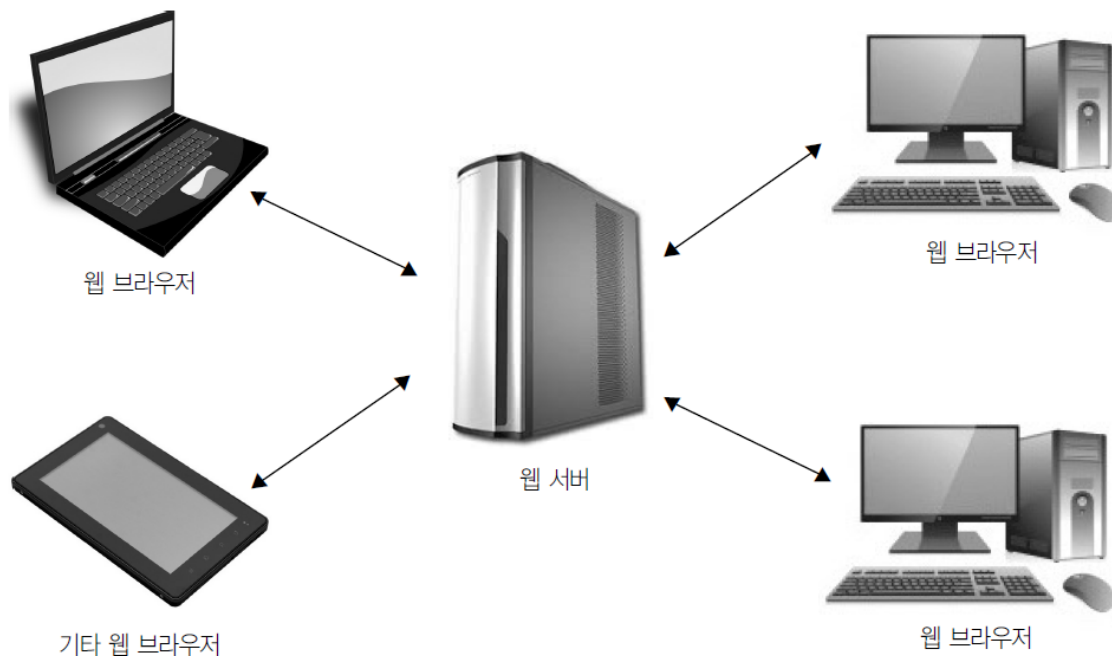
학습내용

- ❖ 웹과 동적 컨텐츠란?
- ❖ JSP란 무엇인가?
- ❖ JSP의 특징
- ❖ Servlet의 기본



웹? 웹! 그리고 동적 콘텐츠!

❖ 웹의 동작



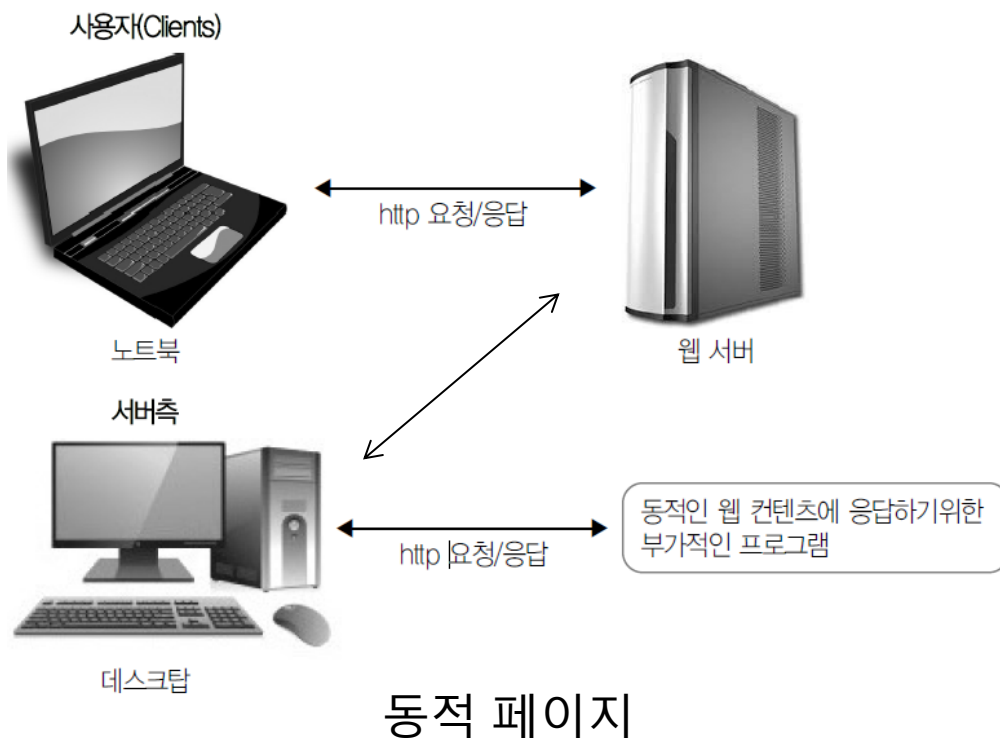
- **요청** : 클라이언트에서 서버로 정보를 요구하기 위해 보내는 메시지입니다. 이 요청 방식에는 GET방식과 POST 방식이 있습니다.
- **응답** : HTTP에서 요구된 메시지에 대한 응답, HTML, 이미지 등이 응답의 내용이 됩니다.



웹? 웹! 그리고 동적 콘텐츠!

❖ 정적 페이지, 동적 페이지

- 정적 페이지 : 변하지 않는 성격을 가진 페이지
- 동적 페이지 : 자동으로 생성된 페이지, 기존의 웹 서버에 새로운 기능을 가진 프로그램을 탑재하여 동적으로 HTML을 생성





웹? 웹! 그리고 동적 콘텐츠!

❖ CGI(Common Gateway Interface)

- 동적 페이지를 위한 기술
- 웹 서버와 동적 콘텐츠 생성을 맡은 프로그램 사이에서 정보를 주고 받는 인터페이스
- 초기 웹 프로그래밍에 사용된 기술
- 프로세스 단위로 실행되기 때문에 사용자 증가 시 급격한 성능 저하





웹? 웹! 그리고 동적 콘텐츠!

❖ 확장 CGI

- 전통적인 CGI 방식의 단점 보완
- 프로세스 생성방식의 변화로 시스템의 부하를 줄임
- ASP, PHP, Servlet, JSP 등

ASP

MS사에서 만들어진
확장 CGI,
특정 웹 서버와 OS에
동작

PHP

ASP와는 달리
특정 영역에서만
동작하지않음,
적은 명령어들로서
프로그래밍이 가능,
기능들이 미약

Servlet

Sun사, Java 언어를
기반으로 하여 동적인
컨텐츠를 생성,
Java 코드안에
HTML태그가
혼재되어 있어서
작업에 대한 분리적인
측면에서 효율성이
떨어짐

JSP

Sun사, Java언어를
기반, HTML태그
사이에 스크립트
언어 형식으로
프로그램을 작성



웹? 웹! 그리고 동적 콘텐츠!

❖ 동적 콘텐츠 처리를 위한 기술들

용 어		설명
CGI		<ul style="list-style-type: none">- 동적 콘텐츠 처리를 위한 가장 전통적인 방법- CGI용 프로그램은 CGI 규약만 지키면 어떤 언어라도 가능- 프로세스 기반 프로그램으로 시스템의 효율성 측면에서 시스템에 많은 부하
확장 CGI	ASP	<ul style="list-style-type: none">- 비주얼 베이직 언어의 문법들중 일부를 사용- ActiveX 컴포넌트로 접근가능- IIS 와 사용(윈도우 운영체제)
	PHP	<ul style="list-style-type: none">- C언어와 비슷한 문법을 사용- 리눅스처럼 소스가 공개되어 있음- 윈도우, 유닉스 운영체제지원, 아파치, IIS 서버 등 지원
	Servlet/ JSP	<ul style="list-style-type: none">- Java 언어를 기반으로 하여 Java 언어의 장점을 수용- 플랫폼에 대해 독립적임과 동시에 컴파일된 파일을 재컴파일 없이 즉시적인 이식이 가능- 사용자정의 태그로 기능 확장이 가능



웹? 웹! 그리고 동적 콘텐츠!

서버측
스크립트

클라이언트측
스크립트

- ✓ 서버측에서 동적인 콘텐츠 생성을 담당하는 기술
- ✓ 서버측에 위치해서 실행되는 스크립트 언어
- ✓ **HTML**과 스크립트 언어를 함께 사용할 수 있는 기술로 웹 서버에서 해석
- ✓ 데이터베이스 연동 처리 등 다양한 구현이 가능
- ✓ 별도의 컴파일 과정없이 **HTML** 태그 수정 가능
- ✓ **JSP, Servlet, PHP, CGI, ASP**

- ✓ 웹 브라우저에서 실행되는 언어
- ✓ 자바 스크립트, **VB** 스크립트



JSP란 무엇인가?

- ❖ 웹 프로그래밍 언어들 중의 하나
- ❖ 동적인 페이지를 생성하기 위한 서버측 스크립트 언어
- ❖ 자바언어를 기반
 - 객체지향적
 - 플랫폼 독립적
 - 네트워크 지향적
 - 뛰어난 보안성
 - 멀티스레드 기능
 - 친근한 코드



JSP란 무엇인가?

❖ 서버 스크립트인 JSP로 구현한 예

```
<%@ page import="java.util.*,java.text.*" %>
<html>
<head><title>example</title></head>
<body>
<%
    Date date = new Date();
    SimpleDateFormat simpleDate = new SimpleDateFormat("yyyy-MM-dd");
    String strdate = simpleDate.format(date);
%>
오늘 날짜는 <%=strdate%>입니다.
</body>
</html>
```



JSP의 특징

빈즈 자바컴포넌트를 사용가능

최초 서블릿컴파일 후 메모리에서 처리
(사용자 접속이 많아도 원활하게 처리)

다른 서블릿간의 간편한 데이터 공유
(page, request, session, application scope)

자바의 모든 기능을 사용가능
(확장성)

IBM, 오라클, 썬, BEA 등에서 강력하게 지원

사용자정의 태그 생성 및
다양한 기능의 태그라이브러리 이용가능(JSTL 등)

다양한 운영체제와 개발/실행환경 지원

JSP의 특징



Servlet의 기본

완전한 자바파일의
형태

JSP와는 달리
자바 코드 안에 **HTML**
태그들이 포함

JSP 페이지는 하나의
서블릿으로 변환이
되고 다시 컴파일 되어
실행

JSP는 브라우저에
표현이 될 부분에 보다
유용하고 서블릿은
브라우저에 표시될
필요가 없지만
내부적으로 처리되어야
할 부분에 보다
유용하게 사용



Servlet의 기본

❖ Servlet으로 구현한 예

```
public class ServletForm extends HttpServlet{
    public void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException{
        PrintWriter out;
        String username = request.getParameter("UserName");
        String userpass = request.getParameter("UserPass");

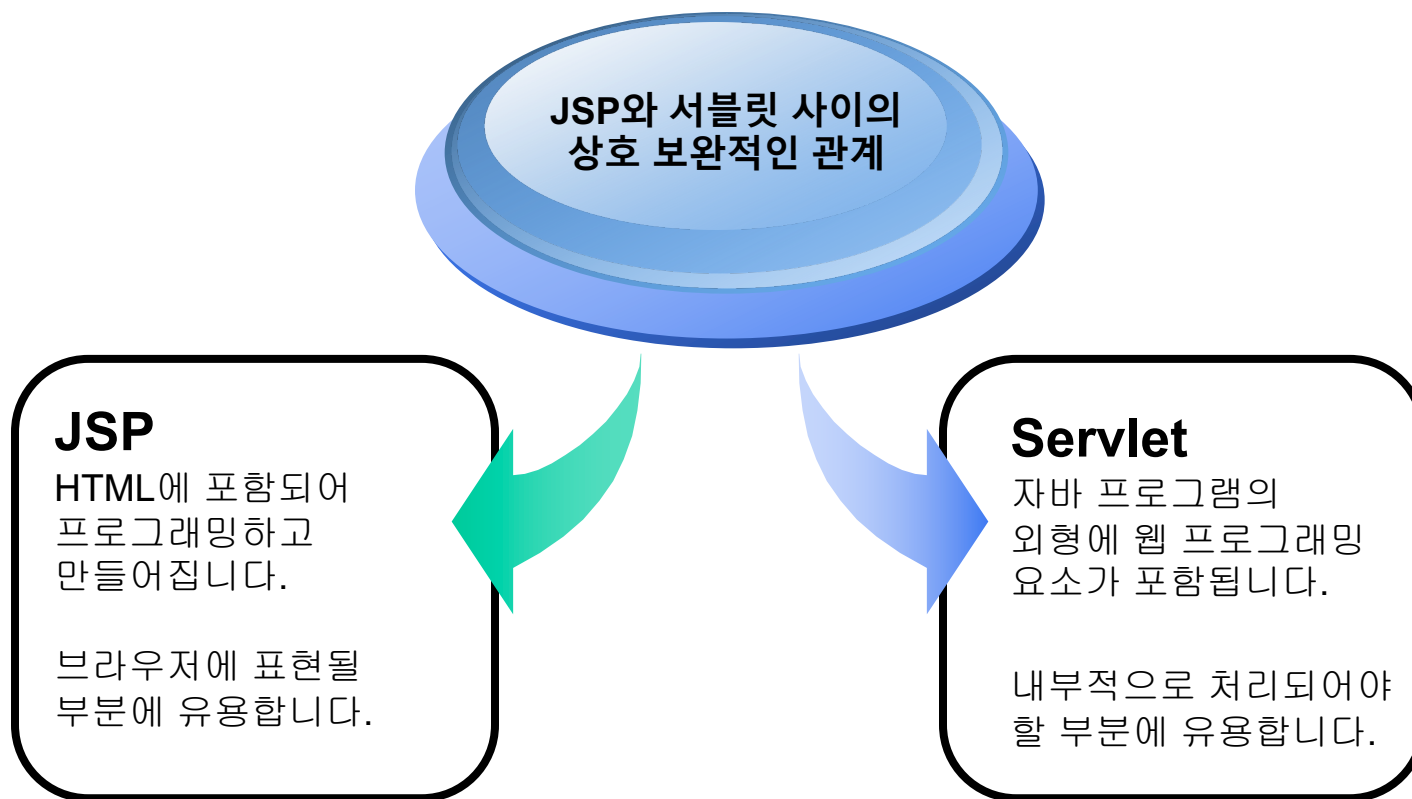
        response.setContentType("text/html;charset=euc-kr");
        out = response.getWriter();

        out.println("<HTML>");
        out.println("<BODY>");
        ...
        out.println("</HTML>");
        out.println("</BODY>");

        out.close();
    }
}
```



Servlet의 기본





Thank You !