



열린사이버대학교  
OPEN CYBER UNIVERSITY

본 파워포인트 디자인은 [열린사이버대학교 저작물]입니다.  
외부 강의사용은 물론 무단적인 복사 및 배포를 금합니다.

# 11. 자바 웹 프로그래밍 1

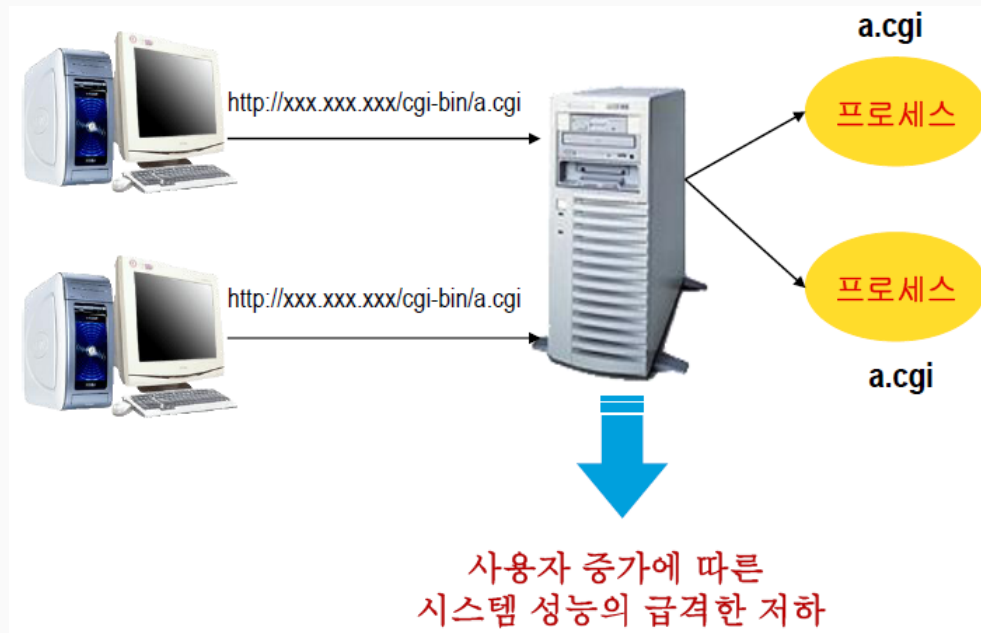
---

11. 자바 웹 프로그래밍 1

## 11. 자바웹프로그래밍1

### 초기 웹프로그래밍 기술(CGI)

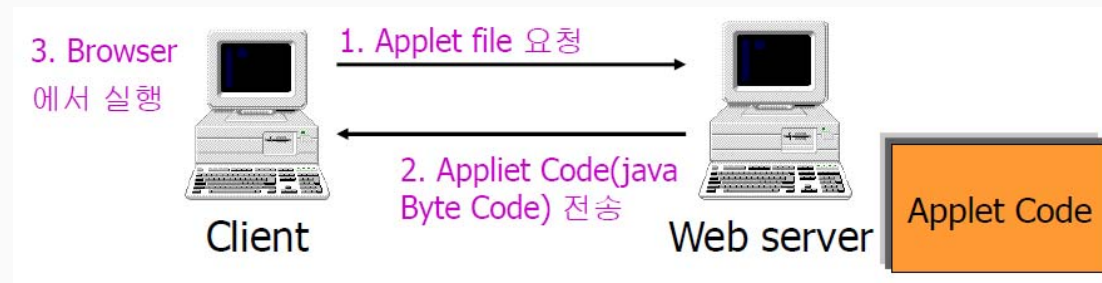
- ❖ 초기웹프로그래밍에사용된기술
- ❖ 프로세스단위로실행되기때문에사용자가증가하면급격히성능



## 11. 자바웹프로그래밍1

### 초기 웹프로그래밍 기술(애플릿)

- ❖ HTML 페이지에 포함되어 자바 호환 웹브라우저에 의해 실행 될 수 있는 자바프로그램
- ❖ Client side 언어



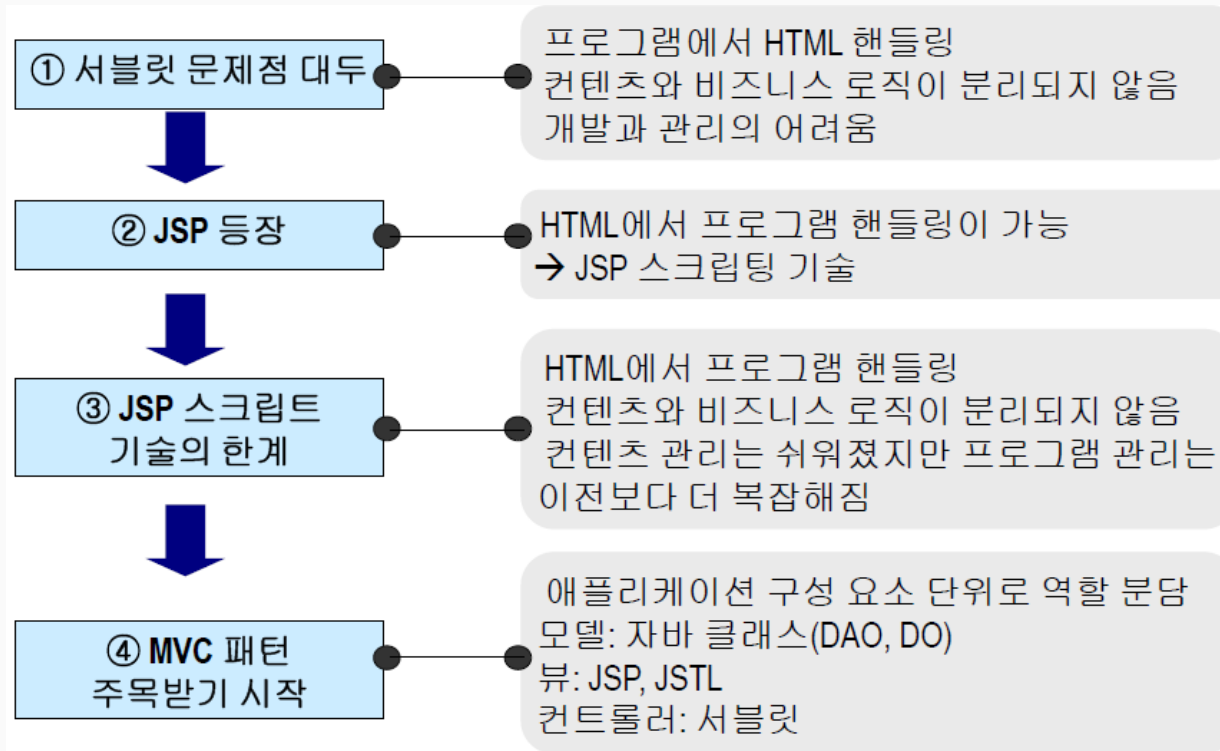
- 사용자는 원하는 애플릿 웹서버에게 요청
  - 웹서버는 사용자가 요청한 애플릿파일을 찾아서 사용자에게 전송
  - 웹서버로부터 받은 애플릿을 브라우저 내에 있는 자바가상머신 상에서 수행
- ❖ 애플릿은 실행할 코드는 서버쪽에 있지만, 실제적으로 이 코드를 실행하는 부분은 클라이언트

### 자바 서블릿 이해하기

#### ❖ 서블릿 개요

- 자바플랫폼에서 컴포넌트기반의 웹 애플리케이션 개발 기술
- JSP는 서블릿 기술에 기반함
- 자바언어를 이용하여 다이나믹한 웹사이트를 개발하기 위해 등장한 기술
- 웹서버의 기능을 확장시키는 방법 제공

### 자바 서블릿 이해하기



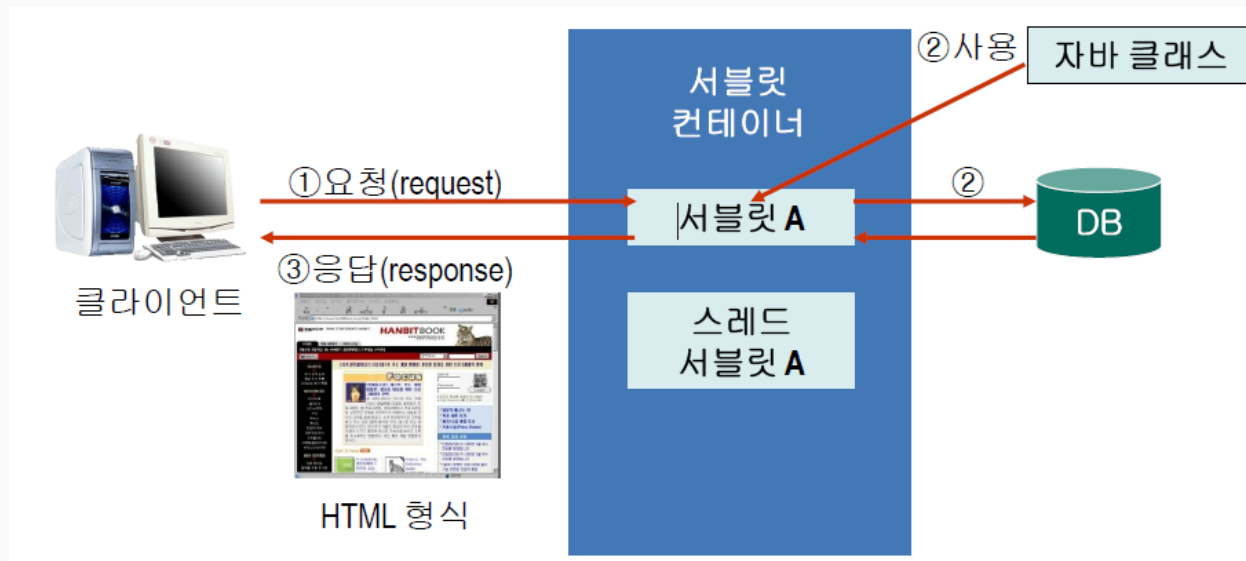
### 자바 서블릿 이해하기

#### ❖ 서블릿 컨테이너(servlet container)

- 자바가상머신을 내장한 서블릿 운영환경.
- JSP는 서블릿으로 변환되어 실행
- 대부분 별도의 실행환경 없이 서블릿 컨테이너에 통합됨.
- 자체 웹서버 기능도 있으나 웹서버와 분리하기도함.
- 대표적으로 apache tomcat이 있음.

### 자바 서블릿 이해하기

#### ❖ 서블릿 동작원리





### 자바 서블릿 이해하기

#### ❖ 서블릿 장점

- 쓰레드를 기반으로 하므로 웹애플리케이션 운영에 효율적이다.
- 자바를 기반으로 하므로 자바 API를 모두 사용할 수 있다.
- 운영체제나 하드웨어에 영향을 받지 않는다.
- 한번 개발된 애플리케이션은 다양한 서버환경에서도 실행이 가능하다.
- 웹애플리케이션에서 효율적인 자료공유방법을 제공한다.

### 자바 서블릿 이해하기

#### ❖ JSP 파일위치

- JSP 파일의 모든 내용은\_jspService() 메서드에 위치

```
public void _jspService(HttpServletRequest request, HttpServletResponse response)
    throws java.io.IOException, ServletException {
    ....
    .....
    out.write("\r\n\r\n");
    out.write("<HTML>\r\n");
    out.write("<HEAD>");
    out.write("<TITLE>Hello World");
    out.write("</TITLE>");
    out.write("</HEAD>\r\n\r\n");
    out.write("<BODY>");
    out.write("<H2>Hello World : 헬로월드");
    out.write("</H2>\r\n오늘의 날짜와 시간은 : ");
    out.print( new java.util.Date() );
    ... 이하중략
}
```

### JSP란

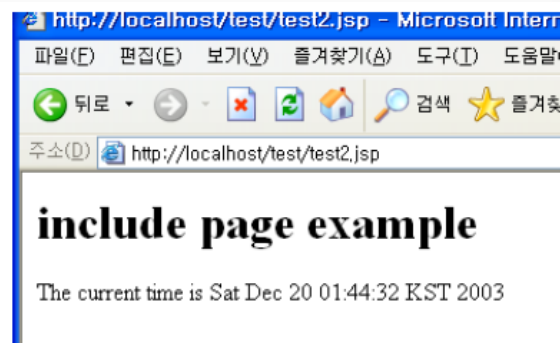
#### ❖ JSP는

- servlet 스펙에기반하여 동적인 web page를 생성하기 위한스펙

#### ❖ Server 측processing

#### ❖ 표현과 동적 콘텐츠를 분리함

```
<%@page import = "java.util.Date"%>
<html>
<body>
<h1>include page example</h1>
The current time is <%= new Date() %>
</body>
</html>
```



### JSP 특징

#### ❖ 특징

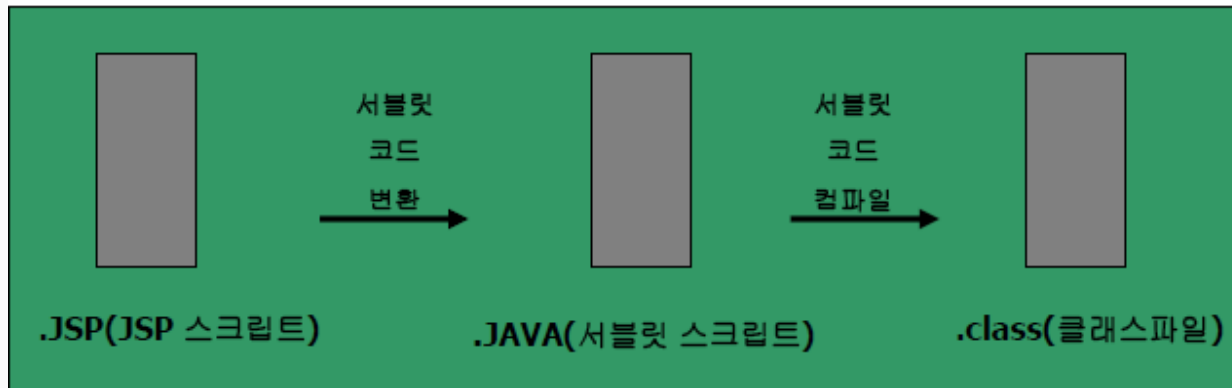
- 보여주는 부분과 프로그램 로직 부분을 분리하여 제작할 수 있다
  - 보여주는 부분: HTML, XML
  - 프로그램 로직: JSP 태그
- 컴포넌트의 재사용
  - 기존에 EJB로 만들어진 컴포넌트가 있다면 JSP에서 그대로 사용가능
  - 프로그램개발(컴포넌트개발)과 웹페이지 개발을 분리하여 작업을 단순화하고 컴포넌트 재사용성을 높임

### JSP 특징

#### ❖ 특징

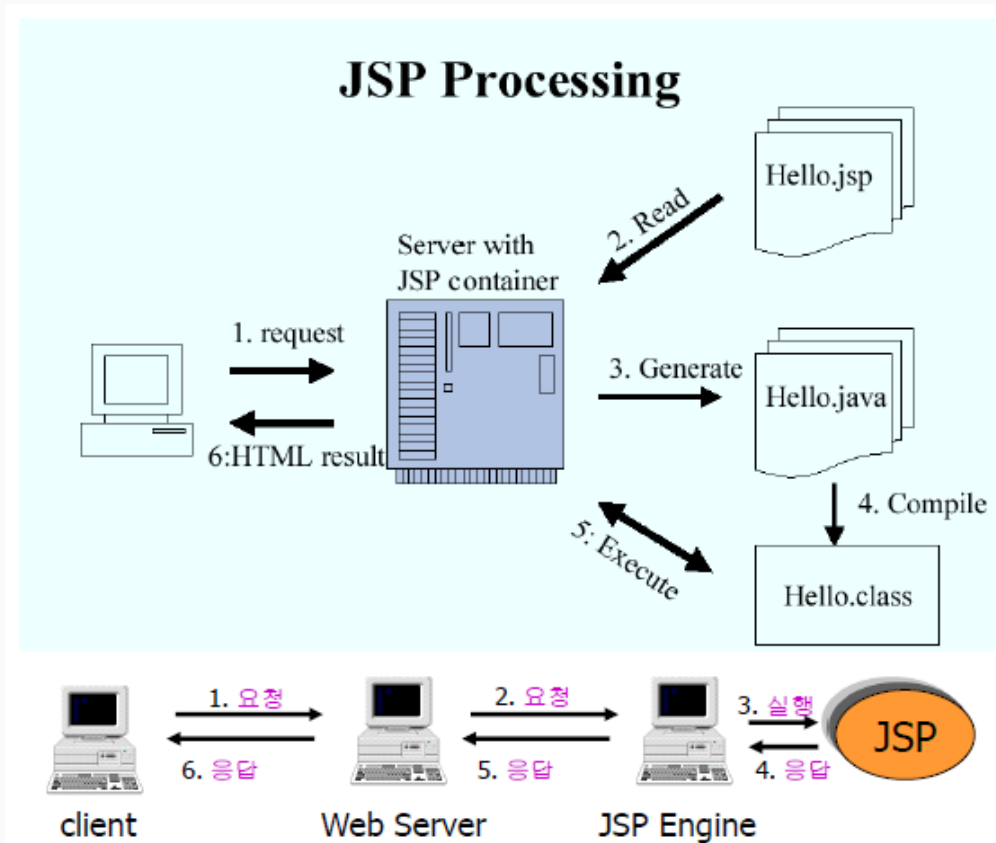
- 자바의 특징 계승
  - JSP는 자바 기반의 스크립트이고, 모든 JSP 페이지는 자바서블릿 코드로 만들어진 후 컴파일 되므로 JSP 페이지는 자바기술이 가진 모든 장점(객체지향프로그래밍, 플랫폼독립성, JDBC의상용등)을 수용
- 페이지의 작성이 쉽다.
  - JSP 파일은 프로그램수행시 코드의 변경유무를 판단해 자동으로 컴파일되므로 서블릿처럼 코드를 작성하고, 다시 이것을 컴파일 해야하는 번거로움이 없음

## JSP 실행과정



- ❖ 사용자가 특정한 JSP 페이지 요청
- ❖ JSP 엔진은 사용자가 요청한 JSP 페이지가 새로 작성되거나 변경되었는지, 아니면 변화되지 않은 것인지를 판단
- ❖ 새로요청(변경) : JSP 파일을 이용해서 블릿코드를 만든 후 이것을 컴파일하여 서블릿을 실행
- ❖ 재요청: 위 과정 생략하고 그대로 전에 만들었던 서블릿을 실행

## JSP 프레임워크



### JSP의 장점

- ❖ 비록 JSP가 servlet이 할 수 없는 일을 할 수는 없지만 JSP는 여러가지 일을 쉽게 할 수 있도록 도와준다.
  - HTML 작성
  - HTML을 읽어들이고 유지관리
- ❖ JSP는 또한 다음과 같은 일을 가능하게 해준다.
  - Macromedia DreamWeaver, 나모웹에디터 등과 같은 표준HTML 툴을 사용해서 코딩가능
  - 팀의 일부 멤버들은 HTML layout 을 하는 것으로 분리가능
- ❖ JSP encourages you to
  - Java code(logic)와 HTML code(표현)을 분리 시킬 수 있다.



### 다른 기술들 (*PHP, ASP*)에 대한 *JSP*의 특징

#### ❖ Versus ASP

#### ❖ 유사점

- 모두 서버측 언어이며, JSP는 ASP에 대한 Java 진영의 구현이다.

#### ❖ 차이점

- ASP
  - 마이크로소프트 프로그래밍언어임(Visual Basic 등과같은)
  - 마이크로소프트의 플랫폼에서 동작
  - ASP는 캐시 되기도 하지만, 대부분의 경우에 항상 인터프리팅 되어 실행됨
- JSP
  - Java Technology
  - 플랫폼 독립적
  - 컴파일된 코드(속도가 빠름)

### 다른기술들 (*PHP, ASP*)에 대한 *JSP*의 특징

- ❖ **Versus client-side JavaScript(in browser)**
- ❖ **JSP와 역할이 중복되지는 않지만 (독립적으로 사용가능)**
  - 클라이언트가 아닌 Server를 제어하는 것임
- ❖ **Java script는클라이언트 측 언어임**
  - 인터프리팅 방식이며, Portable 하지 않음
- ❖ **JSP는 서버측 언어임**
  - 컴파일된 코드
  - Portable함
  - HTML과 integration 되지 않음 – Java가 HTML을 생성함
- ❖ **Versus static HTML**
  - 동적인 측면
  - 웹에서 동적인 측면을 부여하는 것은 더 이상 선택이 아니다.

### 다른기술들 (*PHP, ASP*)에 대한 *JSP*의 특징

#### ❖ Versus Servlets

#### ❖ 유사점

- 서버 측 실행
- End user에게 동일한 결과(identical result)를 제공함
- JSP는 Servlet으로 변환됨

#### ❖ 차이점

- Servlet : "HTML in Java Code"
  - 표현과 로직이 혼합됨
- JSP : "Java Code Scriptlets in HTML"
  - 로직으로부터 표현을 분리함
  - 처음 실행하는 동안에만 느려질 수 있음(컴파일 과정)

## 11. 자바웹프로그래밍1

다른기술들 (*PHP, ASP*)에 대한 *JSP*의 특징

안녕하세요. 오늘은 2003-10-12일 입니다.

```
<%@ page import = "java.util.Date" %>
<%! Date today = new Date(); %>
<html>
<head></head>
<body bgcolor="white" text="black" link="blue" blink="purple"
      alink="red">
안녕하세요. 오늘은 <%=today.getDate() %>일입니다.
</body>
</html>
```

다른기술들 (*PHP, ASP*)에 대한 *JSP*의 특징

```
import java.util.Date;
import javax.servlet.*;
import javax.servlet.http.*;
public class Itisservlet extends HttpServlet
{
    public void service(HttpServletRequest req,
        HttpServletResponse res) throws ServletException, IOException
    {
        Date today = new Date();
        res.setContentType("text/html; charset=euc-kr");
        PrintWriter out = m_response.getWriter();
        out.println("<html>");
        out.println("<head></head>");
        out.println("<body bgcolor=\ "white\ " text=\ "black\ "
            link=\ "blue\ " blink=\ "purple\ " alink=\ "red\ ">");
        out.println("안녕하세요. 오늘은 " + today.getDate() + "일입니다.");
        out.println("</body>");
        out.println("</html>");
        out.close(); } }
```

### JSP 실행환경

- ❖ **Code를 컴파일 할 필요가 없다.**
- ❖ **JSP page를 특별한 디렉토리에 넣을 필요가 없다.**
  - Install\_dir\webapp\ROOT\\*(HTML and JSP-Tomcat)
  - install\_dir\servers\default\default-app (Jrun)
- ❖ **JSP 페이지를 호출하기 위해서 특별한 URL을 사용할 필요가 없다.**
- ❖ **단점**
  - JSP page에 사용되는 일반적인 Java class에 대해서는 CLASSPATH, install dirs, 등에 관한 규칙이 여전히 적용되어야 한다.