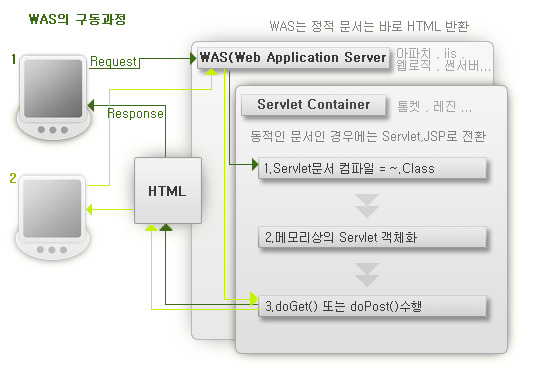
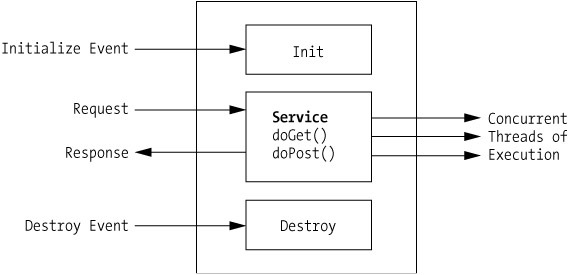
**Servlet**

servlet은 톰캣 위에서 동작하는 java 프로그램입니다. 서블릿 초기화는 다음과 같은 과정을 거칩니다.   


톰캣과 같은 WAS 가 java 파일을 컴파일해서 Class로 만들고 메모리에 올려 Servlet 객체를 만들게 되고 이 Servlet 객체는 doPost, doGet을 통해 요청에 응답합니다. 초기화 과정을 더 자세히 보면   
  
  
  
  
init, Service, destory 과 같은 콜백이 각 시점에 불리는걸 볼 수 있습니다. init은 서블릿이 메모리에 로드 될때 실행됩니다. destory는 마찬가지로 언로드되기 전에 수행되는 콜백입니다.  service 메소드는 HTTP Method 타입에 따라 doGet 혹은 doPost를 호출합니다.   
  
기억해야 할 점은, 초기화된 서블릿이 클라이언트의 요청이 있을 때 마다 Thread를 생성해서 병렬적으로 service를 수행한다는 것입니다. 서블릿 객체는 **여러 개 생성되지 않습니다.** 즉, 서블릿을 요약하면   
WAS 를 통해 컴파일 된 후 메모리에 적재되어 클라이언트의 HTTP Get, Post 등의요청을 처리하는 자바 프로그램.

으로 정리할 수 있습니다.  
Servlet 샘플 코드를 보면, HTML 을 넣기 굉장히 불편합니다.

public class HelloServlet extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException,IOException {

     response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

out.println("<HTML>");

out.println("<BODY>");

out.println("Hello World!!");

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

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

out.close();

}

}

**JSP**

위와 같은 불편한 점은 JSP가 나오게 된 이유입니다. HTML을 코딩하기에 너무 불편해서, HTML 내부에 Java 코드를 삽입하는 형식의 JSP가 등장했습니다.

<%@page import="java.util.Calendar" %>

<%@ page contentType="text/html; charset=UTF-8"%>

<%

String str=String.format("%tF",Calendar.getInstance());

%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01

Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

</head>

<body>

오늘은 <%=str%><br/>

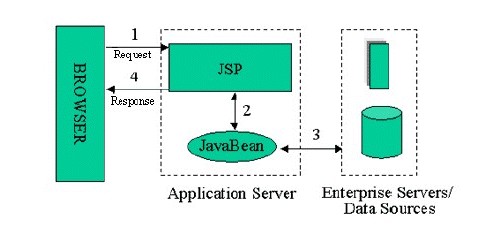
</body>

</html>

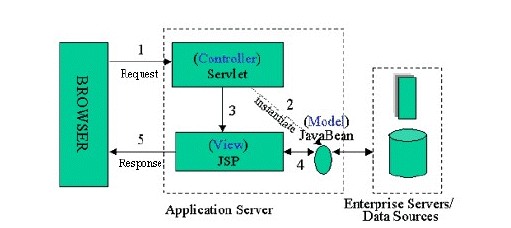
위와 같이 HTML 내부에 Java 코드가 있어 HTML 코드를 작성하기 쉽습니다. 단, **로직과 디자인이 한 파일 내에 섞여**있어 유지보수가 어렵습니다.

결국엔 java 클래스 파일로 변환되어 메모리 상에 적재되지만, 개발자들이 HTML 부분을 편하게 코딩할 수 있게 만든 것이 JSP라고 할 수 있습니다. 그리고 이런 JSP를 이용해서 코딩을 하는 방식에는 크게 두 가지가 있습니다.

**Model 1**

  
사용자로부터 요청을 JSP가 받아(더 정확히는 JSP 에서 사용자가 요청을 합니다.) Java Bean(DTO, DAO)을 호출해 처리합니다. 이런 방식을 Model1 이라고 합니다.   
  
- 개발 속도가 빠르고   
- 배우기 쉽지만   
- 프레젠테이션 로직과 비즈니스 로직이 혼재   
- JSP 코드가 복잡해져 유지 보수가 어려워집니다.

**Model 2**

  
단순히 JSP 만 사용하거나, Servlet만 사용하는 것이 아니라 두 개의 장단점을 모두 취해 View 는 JSP로, Controller 는 Servlet 을 사용한 것이 바로 Model2 입니다. 보여지는 부분은 HTML이 중심이 되는 JSP, 다른 자바 클래스에게 데이터를 넘겨주는 부분은 Java 코드가 중심이 되는Servlet 이 담당하게 됩니다. 그리고 Model 영역 에서는 DTO, DAO를 통해 Mysql 과 같은 Data Storage 에 접근합니다. 