



JSP(Java Server Pages)란

일반적인 웹 페이지는 대부분 미리 서버에 등록해 두었다가 웹 브라우저가 서버에 특정 데이터를 요청하면 이를 웹 브라우저로 전송해서 보여준다. 이를 가능케 하는 기술 중 하나가 JSP(Java Server Pages) 이다.

HTML 내부에 자바(java)코드를 삽입하는 형식이다. 다시 말해 서블릿의 단점을 보완하고자 만든 서블릿 기반의 스크립트 기술이다. 서블릿을 이용하게 되면 웹프로그래밍을 할 수 있지만 화면 인터페이스 구현에 너무 많은 코드를 필요로 하는 등 비효율적인 측면들이 있다. 때문에 서블릿을 작성하지 않고도 간편하게 웹프로그래밍을 구현하게 만든 기술이 JSP(Java Server Pages)이다.

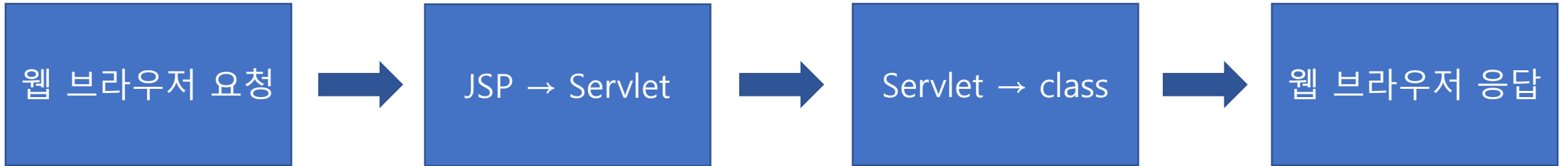


JSP 특징

JSP 특징
동적 웹 어플리케이션 컴포넌트이다.
확장자 → .jsp 이다.
클라이언트 요청에 동적작동, HTML 페이지로 변환되어 응답된다.
서블릿으로 변환되어 실행된다.
MVC 패턴에서 View로 이용된다.



JSP 동작 원리



클라이언트가 *.jsp에 대한 요청을 웹 서버에 하게 된다. 웹 서버는 다시 이 요청을 JSP 컨테이너로 넘긴다. JSP 컨테이너는 해당 JSP 페이지를 찾아 서블릿(자바 파일 /*.java)으로 변환 시킨다. 자바 파일을 컴파일 한다. 자바 파일의 결과가 다시 사용자의 웹 브라우저로 전송되어 클라이언트가 요청한 결과를 받아볼 수 있게 된다.



JSP 태그의 개념 및 종류

Servlet은 JAVA언어를 이용하여 문서를 작성하고, 출력객체(`PrintWriter`)를 이용하여 HTML 코드를 삽입하지만, JSP는 Servlet과 반대로 HTML 코드에 JAVA언어를 삽입하여 동적 문서를 작성 할 수 있다.

HTML 코드안에 JAVA코드를 삽입하기 위해서는 JSP태그를 이용해야 한다.

구분	JSP 태그	설명
스크립트릿	<code><% %></code>	JAVA 코드 삽입
선언문	<code><%! %></code>	변수, 메소드의 선언
표현식	<code><%= %></code>	결과값 출력
지시자	<code><%@ %></code>	페이지 속성 지정
주석	<code><%-- --%></code>	주석 처리
액션태그	<code><jsp:action> </jsp:action></code>	페이지 삽입, 공유, 자바빈 사용 등



JSP 내장 객체

개발자가 객체를 별도로 생성하지 않아도 JSP에서 바로 사용할 수 있는 객체이다. 컨테이너가 JSP를 Servlet으로 변환할 때 자동으로 객체가 생성된다.

내장 객체	설명
request	웹 브라우저의 요청 정보를 저장하고 있는 객체
response	웹 브라우저의 요청에 대한 응답 정보를 저장하는 객체
out	JSP 페이지의 출력할 내용을 가지고 있는 출력 스트림 객체
session	하나의 웹 브라우저 내에서 정보를 유지하기 위한 세션 정보를 저장하고 있는 객체
page	JSP 페이지를 구현한 자바 클래스 객체
config	JSP 페이지에 대한 설정 정보를 담고 있는 객체
exception	JSP 페이지에서 예외가 발생한 경우 사용하는 객체

Request 내장 객체

웹 브라우저를 통해 서버에 어떤 정보를 요청하는 것을 request 라고한다. 이러한 요청정보는 request 객체가 관리한다. (웹 브라우저에서 전달되는 정보의 모임)

form에 입력한 요구 사항을 얻어낼 수 있도록하는 메소드 제공

메소드	설명
getParameter(name)	파라미터 변수 name에 저장된 값을 얻어내는 메소드 해당 변수명이 없으면 null 리턴 (단독값을 입력하는 text, select, radio 등에 사용)
getParameterValues(name)	파라미터 변수 name에 저장된 모든 값을 얻어내는 메소드 변수값은 String 배열로 리턴 (다중값을 입력하는 checkbox 등에 사용)
getPrameterNames()	요청에 의해 넘어오는 모든 파라미터 이름들을 java.util.Enumeration 타입으로 리턴



Request 내장 객체

request 객체는 웹 브라우저와 웹 서버의 정보도 가져올 수 있다.

메소드	설명
getServerName()	서버 이름을 얻는다.
getContextPath()	해당 JSP페이지가 속한 웹 애플리케이션의 컨텍스트 패스를 얻는다.
getMethod()	요청에 사용된 요청 방식(GET, POST 등)을 얻는다.
getSession()	세션 객체를 얻는다.
getProtocol()	해당 프로토콜 객체를 얻는다.
getRequestURL()	요청 URL를 얻는다.
getRequestURI()	요청 URL를 얻는다.



Response 내장 객체

response 객체는 JSP 페이지의 실행결과를 웹 브라우저로 되돌려 줄 때 사용되는 객체이다. 서버에서 클라이언트에 대한 응답을 response 객체를 통하여 처리한다.

메소드	설명
getCharacterEncoding()	응답할 때 문자의 인코딩 형태를 구한다.
addCookie(Cookie)	쿠키를 지정한다
sendRedirect(URL)	지정한 URL로 이동한다. 특정상황에 조건에 따라 원하는 페이지로 이동하게 한다.(로그인 시 비회원이면 회원가입페이지로...)



JSP 액션태그

액션태그란 JSP 페이지 내에서 어떤 동작을 하도록 지시하는 태그 이다.
페이지 이동, 페이지 include, 자바 빈 생성 등이 있다

태그	설명
forward	현재 페이지를 다른 페이지로 전환할 때 사용한다.
include	현재 페이지에 다른 페이지를 삽입하는 액션태그이다.
param	forward 및 include 태그에 데이터 전달을 목적으로 사용되는 태그이다. 이름과 값으로 이루어져 있다.



forward와 redirect

forward 방식 : 웹 컨테이너 차원의 페이지 이동, 웹 브라우저는 다른 페이지로 이동했음을 알 수 없고, 브라우저에 최초 호출한 URL이 표시된다.

redirect 방식: 다른 웹 컨테이너에 있는 주소로 이동 가능, 웹 브라우저는 URL을 지시된 주소로 바꾸고 그 주소로 이동

forward 방식	sendRedirect 방식
JSP 액션 태그이다.	JSP 내장 객체이다. (response)
요청 정보가 유지된다.	새로운 요청이 수행된다.
객체를 재사용한다.	객체를 재사용하지 않는다. (새로운 요청)
시스템 변화가 생기지않는 요청에 적합 (검색 등의 단순 조회)	시스템 변화가 생기는 요청에 적합 (로그인, 글쓰기 등)



include

include란 현재의 문서에 다른 문서, 즉 다른 파일의 내용을 포함시켜 출력하는 것을 말한다.

`<%@ include %>` 지시자 : 원래 페이지 안으로 include 지시어로 지정한 페이지의 소스가 그대로 복사 된다.

`<jsp:include>` 액션태그 : 소스 코드가 복사되는 것이 아니라 제어권 자체가 include 액션으로 지정된 페이지로 넘어갔다가 다시 원래 페이지로 돌아온다.



page / pageContext

JSP(JavaServer Pages)에서 pageContext 객체와 page 객체는 둘 다 JSP 페이지에서 사용할 수 있는 객체이다.

pageContext : JSP 페이지와 관련된 정보를 포함하고 있는 컨테이너 역할을 한다. 주로 스코프 접근, 페이지 설정 및 페이지 간 데이터 전달과 관련기능을 제공한다.

page 객체 : 현재 JSP 페이지 자체를 나타내는 객체로, JSP 페이지 내에서 자바 코드를 실행하고 서버 측 로직을 구현하는 데 사용된다.



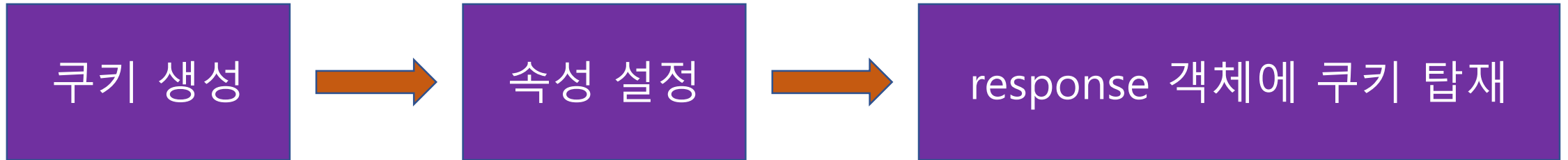
쿠키란?

웹브라우저에서 서버로 어떤 데이터를 요청 하면, 서버측에서는 로직을 수행한 후 웹브라우저에 응답한다. 그리고 서버는 웹브라우저와의 관계를 종료한다. 이렇게 웹브라우저에게 응답 후 관계를 끊는 것은 http 프로토콜 특징이다.

연결이 끊겼을 때 어떤 정보를 지속적으로 유지하기 위한 수단으로 쿠키라는 방식을 사용한다. 쿠키는 서버에서 생성해서 클라이언트측에 특정 정보를 저장한다.

쿠키는 4kb로 용량이 제한적이며, 300개까지(1.2MB) 데이터 정보를 가질 수 있다.

쿠키 과정

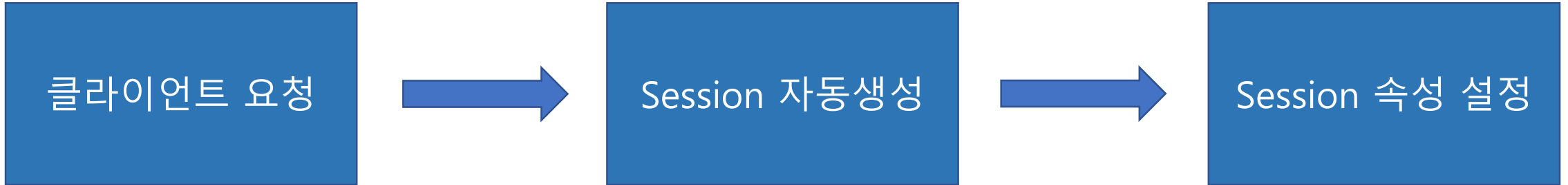


1. javax.servlet.http 패키지에 있는 Cookie 객체를 생성 한다.
2. 생성된 쿠키객체에 속성설정을 한다.
3. response 객체에 실려서 사용자의 웹 브라우저에 저장한다.

쿠키 관련 메소드

메소드	설명
setMaxAge()	쿠키의 유효한 기간을 설정한다.
setPath()	쿠키사용의 유효한 디렉토리를 설정한다.
setValue()	쿠키의 값을 설정한다.
getMaxAge()	쿠키 유효기간 정보를 가져온다.
getPath()	쿠키의 유효한 디렉토리 정보를 가져온다.
getName()	쿠키의 이름을 가져온다.
getValue()	쿠키에 설정된 값을 가져온다.

세션이란?



세션도 쿠키와 마찬가지로 서버와의 관계를 유지하기 위한 수단이다.
단, 쿠키와 달리 클라이언트의 특정 위치에 저장되는 것이 아니라, 서버상에 객체로 존재한다. 따라서 세션은 서버에서만 접근이 가능하여 보안이 좋고, 저장할 수 있는 데이터에 한계가 없다.

세션 관련 메소드

메소드	설명
<code>setAttribute(String name, Object value)</code>	세션 속성명이 name인 속성에 속성값으로 value를 할당한다.
<code>getAttribute(String name)</code>	세션 속성명이 name인 속성의 값을 Object 타입으로 리턴한다. 해당 되는 속성명이 없을 경우에는 null 값을 리턴한다.
<code>getAttributeNames()</code>	세션 속성의 이름들을 Enumeration 객체 타입으로 리턴한다.
<code>getId()</code>	세션에 할당된 고유 식별자를 String 타입으로 리턴한다.

세션 관련 메소드

메소드	설명
<code>setMaxInactiveInterval(int interval)</code>	세션 유지시간을 초 단위로 설정한다.
<code>getMaxInactiveInterval()</code>	세션의 유효시간을 초 단위로 얻는다.
<code>removeAttribute(String name)</code>	세션 속성명이 name인 객체를 삭제한다.
<code>invalidate()</code>	세션의 모든 데이터를 삭제한다.

어플리케이션이란?

웹 어플리케이션에 대한 정보를 저장한다. 웹 브라우저가 종료되면 데이터가 초기화되는 세션(session)과는 달리, 어플리케이션(application)은 서버 자체가 종료되기 전까지 유지된다.

메소드	설명
setAttribute(String name, Object Value)	어플리케이션에 값을 저장
getAttribute(String name)	setAttribute()로 저장할때 사용했던 변수값으로 어플리케이션에 저장된 데이터를 불러옴
removeAttribute(String name)	해당 변수명으로 저장된 어플리케이션값을 삭제

예외페이지

JAVA에서 예외처리 (try-catch)를 사용하듯이, JSP에서도 예외가 발생할 수 있다. 예외적 상황이 발생했을 경우, 톰캣에서 제공되는 예외 페이지가 보여지면 사용자 입장에서 불쾌감이나 거부감이 들 수 있다. 따라서 서버에서 제공되는 기본적인 에러 페이지보다 예외처리 페이지를 직접 만들고 예외페이지를 지정한다.

HTTP 상태 404 – 찾을 수 없음

타입 상태 보고

메시지 파일 [/helloJS.jsp]을(를) 찾을 수 없습니다.

설명 Origin 서버가 대상 리소스를 위한 현재의 representation을 찾지 못했거나, 그것이 존재하는지를 밝히려 하지 않습니다.

Apache Tomcat/9.0.50



page 지시자를 이용

에러 시 `errorPage.jsp`로 이동

```
<%@ page errorPage="errorPage.jsp"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<%
    int i = 10/0;
%>
```

200 : 요청을 정상 처리

```
<%@ page isErrorPage="true"%>
<% response.setStatus(200); %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    에러 발생<br />
    <%= exception.getMessage() %>
```



web.xml 파일을 이용

HTTP는 여러 에러 코드들이 있다. 이 중에서 404와 500이 대표적인 JSP 에러이다. web.xml 파일을 이용해 에러페이지를 처리해 줄 수 있다.

404 : 요청한 페이지가 없을 때

500 : 서버에서 자바를 실행하다가 에러가 났을때

```
<error-page>
  <error-code>404</error-code>
  <location>/step05/error404_page.jsp</location>
</error-page>

  <error-page>
    <error-code>500</error-code>
    <location>/step05/error500_page.jsp</location>
  </error-page>
```



자바 빈

JSP에서 여러 데이터 작업을 효율적으로 사용하기 위해 빈(bean)을 사용한다.

JAVA 언어의 속성(필드)과 기능(메소드)으로 이루어진 클래스를 말한다.

JSP에서는 속성과 기능을 가진 클래스를 빈(bean) 이라고 한다.

빈을 만든다는 것은 데이터 객체를 만들기 위해 클래스를 만드는 것이다.

데이터 객체에는 데이터가 있어 그에 해당하는 (getter / setter) 메소드가 있다.

사용방법

1. 속성과 기능이 있는 빈 클래스(데이터 클래스)를 만든다.
2. jsp 페이지를 만들어서 액션태그를 이용하여 데이터 클래스를 사용한다.
(빈의 내부 데이터를 처리한다.)



빈 만들기

```
public class Student {  
  
    private int studentID;  
    private String name;  
    private int age;  
    private int grade;  
  
    public Student() {}  
}
```

```
public int getStudentID() {  
    return studentID;  
}  
  
public void setStudentID(int studentID) {  
    this.studentID = studentID;  
}  
  
public String getName() {  
    return name;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
}
```


빈 관련 액션태그

useBean : 특정 Bean을 사용한다고 명시 할 때 사용한다.

```
<jsp:useBean id="student" class="com.java.jsp.Student" scope="page" />
```

속성	설명
id	자바 객체의 이름을 명시
class	자바빈 클래스(패키지명을 포함)를 기술
scope	자바빈 객체의 유효 범위로 자바빈 객체가 공유되는 범위를 지정

scope	
page	생성된 페이지 내에서만 사용 가능하다.
request	request요청을 받고 처리를 완료할 때까지 유지된다.
session	클라이언트당 하나씩 할당된다. 브라우저를 종료하기 전까지 유지된다.
application	사이트 전체의 범위, 서버가 종료되기 전까지 계속 유지된다.



빈 관련 액션태그

setProperty : 데이터 값을 설정 할 때 사용한다.

```
<jsp:setProperty name="student" property="name" value="홍길동"/>  
<jsp:setProperty name="student" property="age" value="20"/>
```

속성	설명
name	빈 이름
property	속성 이름
value	속성(데이터) 값



빈 관련 액션태그

getProperty : 데이터 값을 가져올 때 사용한다.

```
<jsp:getProperty name="student" property="name"/><br/>  
<jsp:getProperty name="student" property="age" /><br/>
```

속성	설명
name	빈 이름
property	속성 이름