Servlet & JSP
Part5-JSP개요

백 성 애

1. JSP란?

- 1...Java Server Page의 약자.
- Java의 웹서버 프로그램 스펙(서블릿)으로 변환되어
- 서비스된다.
- 2. 장점
- 1) 서블릿보다 쉽고, 빠르다.
- 2) 디자인부분과 로직 부분을 분리시킬 수 있다.
- 3) 프로그래머가 직접 코딩한 Servlet보다 최적화된
- Servlet을 생성시켜 효율적인 코드가 만들어진다.
- 4) JavaBeans의 사용이 쉽다.
- 5) 웹애플리케이션 상에서 변수의 유효범위(scope)
- 설정이 쉽다.

2. JSP 구성 요소와 내장 객체

태그	기 능		
HTML태그	HTML 태그나 HTML 문서 내용들은 고정되어 있기 때문에 템플릿 데이터(FIXED TMPLETE DATA)라고 한다.		
	지시어 원소 (Drecitve Element)	● 지시어는 JSP에 관한 정보를 JSP 컨테이너에 전달하는 역할을 한다. ● Page, include, talib 지시어가 존재한다. 예) <%@ page language=" java" contentType=" text/html;charset=euc-kr import=" java.io.*" " %>	
	스크립트 원소 (Script Element) => JSP 페이지에 자바 코드 기술	선언	● JSP페이지의 멤버필드나 멤버메소드를 선언할 경우 예) <%! String str=" hello" %>
		실행 (스크립트렛)	● 자바 코드를 자유롭게 기술할 수 있는 영역 예) <% 마음대로 자바 코드…%>
		출력	● 변수의 값이나 계산결과 함수 호출 결과를 템플릿 데이터로 출력하고자 할 경우 사용. 즉 문자열 형태로 웹브라우저에 전달되어 사용자 화면에 나타난다.
	액션 원소 (Action Element)	● JSP 액션 태그라고 불리는데, 어떤 기능이 특별히 정의돼 있는 태그로서 <jsp:useBean>,<jsp:setProperty>,<jsp:include>,<jsp:forward> <jsp:plugin> 등과 같이 표준 액션 태그들이 있으며, 프로그래머가 직접 생성하여 사용할 수도 있다.(커스텀태그)	

2. JSP 구성 요소와 내장 객체

내장객체	주요 역할	타입	범위
request	사용자로부터 요청을 처리하는 객체	HttpServletRequest	request
response	서버가 사용자에게 처리결과를 응답하는 객체	HttpServletResponse	page
pageContext	JSP 실행에 대한 context 정보를 담고 있는 객체	PageContext	page
session	클라이언트(브라우저)별로 세션정보를 처리하는 객체	HttpSession	session
application	웹서버의 어플리케이션을 처리하는 객체	ServletContext	application
out	사용자에게 응답을 위한 아웃풋스트림을 처리하는 객체	JspWriter	page
exception	예외 발생시 처리하는 객체	Throwable	page
page	현재 jsp페이지에 대한 정보를 담고있는 객체	HttpJspPage	page
config	현재 jsp 페이지에 대한 초기화 환경을 처리하는 객체	ServletConfig	page

2. JSP 구성 요소

- 1) 지시어 요소(Directive Element)
- ex] <%@ page %>

- 2)스크립팅 요소(Scripting Element)
- [1]선언(declaration) <%!자바코드%>
- [2]스크립트렛(scriptlet) <%자바코드%>
- [3]식(expression)<%=자바코드%>

- 3) JSP액션 요소(JSP표준 태그)
- <jsp:태그명></jsp:태그명>

3.지시어 (Directive Element)종류

- 1) 지시어 형태: <%@지시어%>
- 2)종류
- **[1]page 지시어**
- -contentType : "text/html;charset=euc-kr"
- -import : "java.util.*, java.io.*"
- -info :"page지시어에 대한 exampleJSP파일"
- -language : "java"
- -session : "true"(기본값)| "false"
- -buffer : "8kb"(기본값)| "none"
- -autoflush: "true"(기본값)| "false"
- -errorPage: "/error.jsp"
- -isErrorPage: "false"(기본값)| "true"

3.지시어 (Directive Element)종류

- -pageENCODING: "ISO-8859-1"| "euc-kr"
- -extends : JSP엔진에 의해 자동으로 설정
- -isThreadSafe: "true"(기본값)| "false“

- **[2]include 지시어**
- -file : "include.jsp" [페이지 소스를 포함]
- **[3]taglib 지시어**
- -uri : "WEB-INF/tlds/my.tld"
- -prefix: "mytag"

4. 스크립팅 원소(Scripting Element)

- -기능: jsp페이지에서 자바 코드를 직접 기술할 수 있게 하는 기능
- -종류와 형태
- **[1]선언(declaration)** <%!자바코드%>
- **[2]스크립트렛(scriptlet)** <%자바코드%>
- **[3]식(expression)**<%=자바코드%>
- cf>
- **<%!** int a=10; //멤버변수**%>**
- **<%** int b=20; //지역변수
- /*스크립트렛 태그에 구현된 자바 코드는 모두 serviece()메소드 안에 구현된다.따라서 여기에 선언된 변수는 지역변수가 될 수 밖에 없다. 또한 스크립트렛 태그 안에서는 메소드를 구성할 수 없다.*/
- **%>**

5. JSP액션(JSP표준 태그)

- 1> 정의: JSP에서 표준으로 정의한 태그
- 2> 형태: <jsp:태그명/>
- 3> 종류
- **[1] useBean**
- -class : "my.upload.MultipartRequest"
- -id : "mr"
- -scope : page/request/session/application
- **[2] setProperty**
- -name : "mr"
- -property: "myfile"
- -value :"test.txt"
- [mr.setMyfile("test.txt")와 동일한 효과]

5. JSP액션(JSP표준 태그)

- **[3] getProperty**
- -name : "mr"
- -property: "myfile"
- [mr.getMyfile() 과 동일한 효과]
- **[4] include**
- -page : include.jsp [실행결과물을 포함]
- **[5] forward**
- -page : "/result.jsp"
- [페이지를 이동시킴. sendRedirect()와는 약간의 차이가 있다.//return문 쓸 필요가 없음]
- **[6] param**
- **[7] plugin** :애플릿을 플러그인할 경우...등
- **[8] params** : <jsp:params></jsp:params>

6. JavaBeans

- **1>Beeans 정의:**
- [1] 넓은 의미: 자바 컴포넌트(기능 덩어리)
- [2] 좁은 의미: 자바 컴포넌트 중에서 Beans 규약에
- 맞는 컴포넌트
- **2>Beans 사용의 장점**
- [1] 유효범위(scope) 설정을 쉽게 할 수 있다.
- [2] html로부터 넘어오는 데이터를 쉽게
- setting해서 쓸 수 있다.
- [3] 기능을 모듈화시킬 수 있다.-재사용이 쉽다.
- [4] DB 또는 html로부터 넘어온 데이터를
- 객체화시켜서 메모리에 올려놓을 수 있다.
- (DTO객체 또는 Value Object)

6. JavaBeans – 빈즈 규약

- [5] 프리젠테이션(View)과 비즈니스로직(업무 처리로직)을 준독립시켜서 유지 보수가 쉽다.

- **3>빈즈 규약**
- [1] 기본생성자가 존재해야 한다.
- [2] 속성을 지정(private)해야 하는데 속성 이름은 html의 form 태그안에서 input 태그의 name속성값과 같아야 한다.
- [3] 속성을 접근하고 꺼내올 수 있는 setXXX/getXXX메소드를 구성한다.

6. JavaBeans – 빈즈의 종류

- **4> 빈즈의 종류**
- [1] 비쥬얼 컴포넌트 빈즈
- – JButton, JTextField...
- [2] 데이터빈즈
- –데이터를 담아두는 객체를 만드는 클래스
- – StudentDTO, Person, Friend ...
- [3] 서비스 빈즈
- –연산이나 서비스기능 Beans 즉, Worker빈즈
- – StudentDAO, FriendApp...등

6. JavaBeans – 빈즈의 유효범위

- 5> 빈즈의 유효범위
- [1] page : 하나의 jsp 페이지 내에서만 유효
- [2] request: 하나의 요청에 대한 처리관련
- jsp페이지들에 유효
- [3] session(공간): 하나의 사용자(브라우저) 에 대한 유효

** 세션 유효시간**********
-특정 사용자가 웹브라우저를 닫을 때까지
-30분 동안 요청이 없을 때까지 (conf/web.xml에서 세션 시간을 조정할 수 있다.)

- [4] application: 하나의 웹 애플리케이션 전체기간 유효(서버 종료시까지)
- cf)크기 **page <request <session <application**

7. JSP에서 페이지 이동

- <1>**response.sendRedirect("/count3.jsp");**
- -sendRedirect()는 지정된 URL을 request 한다.
- -sendRedirect()이후로도 로직을 계속 수행한다.
- <2> **액션에서 forward로 이동**
- - forward는 redirect와 달리 해당 요청을 서버의
- 다른 자원에 전달한다. 해당 요청을 사용할
- 다음 자원에 전송함
- - forward 이후로 로직을 계속 수행하진 않는다.
- - forward는 클라이언트와 통신 없이 서버에서만 처리되기 때문에 리다이렉트 보다 나은 성능을 보여줌

7. JSP에서 페이지 이동

- <3> 자바스크립트에서
- **location.href("/count3.jsp")**로 이동할 경우
- -지정된 URL로 이동한다.
- (즉 request한다)
- -href()이후로도 로직을 수행한다.
- sendRedirect()와 같음
- 따라서 더이상 로직을 수행 못하게 하려면 return해야 함

8. include 디렉티브와 include 액션의 차이점

- 1) include디렉티브
- <%@ include file=""%>
- 예제] index.jsp [top.jsp,bottom.jsp, left.jsp]

- 2) include 액션
- <jsp:include page="" />
- 예제] memberFind.jsp[result.jsp]

8. include 디렉티브와 include 액션의 차이점

- – include 디렉티브도 <jsp:include>와 마찬가지로 지정한 페이지를 현재 위치에 포함시켜 주는 기능을 제공한다. 하지만 <jsp:include>와 달리 include디렉티브는 실행방식에 있어서 큰 차이를 보인다.

[1]<jsp:include page="">는 **다른 jsp로 흐름을 이동시켜 그 결과물을 현재 위치에 포함시키는** 방식인 반면,

[2]<%@ include file="">%>는 **다른 파일의 내용을 현재 위치에 삽입시킨 후에 jsp파일을 자바 파일로 변환하고 컴파일 하는 방식**인 것이다.

9. COOKIE

- 1) 쿠키란?
- – 쿠키는 클라이언트(웹브라우저)에 저장되는 간단한 정보를 의미.
- – 쿠키는 웹서버와 웹브라우저 양쪽에서 생성할 수 있으며,
- – 웹서버는 웹브라우저가 전송한 쿠키를 사용해 필요한 데이터를 읽어올 수 있다.
- – 쿠키는 그 크기가 하나에 4KB 이하로 제한이 되어 있으며, 총 300개까지 정보를 저장할 수 있다.
- 따라서,최대로 저장가능한 쿠키의 용량은 1200KB 즉 1.2MB

9. COOKIE-쿠키 종류

- 1> 하드에 저장되는 쿠키
- 2> 클라이언트의 메모리에 존재하는 쿠키
- - JSessionId(브라우져가 켜져있을 때는 항상
- 존재)

9. COOKIE-쿠키 동작 방식

- 1> 클라이언트가 처음 서버에 접근하면 서버는 서블릿을 통해 클에 쿠키를 설정한다.

- 2> 후에 사용자가 그 사이트를 다시 방문하는
- 경우 웹브라우저는 쿠키 정보를 서버에 전달한다.

- 3> 서버는 이 쿠키 정보를 이용해 적절한
- 작업을 수행한다.

9. COOKIE-쿠키 설정 절차

- **1) 쿠키 생성 단계**

- 먼저 쿠키를 생성-서블릿/JSP에서 쿠키는
- 웹서버 측에서 생성한다.
- 이렇게 생성된 쿠키는 응답 데이터에 함께
- 저장되어 전송 된다.
- ex) COOKIE ck=new COOKIE("id","swan");

9. COOKIE-쿠키 설정 절차

- **2) 쿠키 속성 부여 및 저장 단계**
- 웹브라우저는 응답 데이터에 포함된 쿠키를 쿠키 저장소에 보관한다.
- 쿠키 종류에 따라 메모리나 파일로 저장된다.
- ex) response.addCOOKIE(ck);

- **3) 쿠키 전송 단계**
- 웹브라우저는 한번 저장된 쿠키를 매번 요청이
- 있을 때마다 웹서버에 전송한다.
- 웹서버는 웹브라우저가 전송한 쿠키를
- 사용해서 필요한 작업을 수행한다.

9. COOKIE-쿠키의 활용

- 일단 웹브라우저에 쿠키가 저장되면, 쿠키가
- 삭제되기 전까지는 매번 웹 서버에 전송된다.
- 따라서 웹어플리케이션을 사용하는 동안
- 지속적으로 유지해야 하는 정보는 쿠키를 사용해
- 저장하면 된다.

- **cf) 쿠키를 이용한 아이디 저장하기**
- … 로그인 하는 사이트를 보면 아이디 기억하기
- 기능을 제공하는 경우가 있다.

9. COOKIE-쿠키의 활용

- 아이디 기억하기 기능은 쿠키를 사용해 구현할 수 있다.
- 사용자가 로그인에 성공하면 아이디를 값으로 저장하고 있는 쿠키의 유효시간을 한달 정도로 여유롭게 잡아서 생성한다.
- 그러면 웹브라우저를 닫아도 유효시간이 충분하기
- 때문에, 다음 브라우저를 열 때 아이디를 저장하고
- 있는 쿠키를 사용할 수 있다.
- 따라서 웹프로그램은 아이디 쿠키가 존재할 경우
- 쿠키의 값을 로그인 폼에 출력해주면
- 암호 기억하기 기능이 구현된다.

9. COOKIE-쿠키의 구성 요소

- - 형태
- :웹서버가 웹브라우저에 전달하는 쿠키를 설정할 때
- **Set-COOKIE : name=value ; expire=date ; path=path; domain=domain; secure**

- 1> 이름 : 각 쿠키를 구별하는 데 사용되는 이름
- 2> 값 : 쿠키의 이름과 관련된 값
- 3> 유효기간 : 쿠키 유효 기간.
- expire을 기술하지 않으면 현재
- 웹브라우저가 실행되는 동안만 유효함
- 4> 도메인: 쿠키를 전송할 도메인
- 5> 경로: 쿠키를 전송할 요청경로

9. COOKIE-쿠키 이름 및 값에 대한 규칙(RFC2109규약)

- 1> 쿠키 이름은 아스키코드 알파벳과 숫자만 포함할 수 있다.
- 2> 콤마(,), 세미콜론(;), 공백(") 등의 문자는 포함할 수 없다.
- 3> $로 시작할 수 없다.
- ...따라서 위와 같은 문자들이 포함되도록 하기
- 위해서는 반드시 인코딩 처리를 해줘야 한다.

- ex) COOKIE c=new COOKIE("name",
- java.net.URLEncoder.encode("백성애"));
- ==> 대신 꺼내올 때는
- URLDecoder.decode(cks[i].getValue());로 꺼내야 함.

9. COOKIE-쿠키보다 세션을 사용하는 이유

- ...세션이 쿠키보다 보안에 앞서기 때문.
- 쿠키 이름과 데이터는 네트워크를 따라 전달되기
- 때문에일반적인 http프로토콜을 사용할 경우
- 중간에 누군가 쿠키의 값을 읽어올 수 있다.
-
- 하지만 세션은 서버에만 저장되기 때문에
- 중요한 데이터를 저장하기에 알맞다.
- 또한 브라우저가 쿠키를 지원하지 않을 경우
- 세션을 사용.

10. 파일 업로드 컴포넌트 다운로드

- **MultipartRequest 를 사용해보자.**
- 다운로드: **http://www.servlets.com/cos/**

- 1. cos-26Dec2008.zip파일을 다운받아 압축풀기
- 2. cos-26Dec2008/lib/안에 **cos.jar** 파일을 복사하여
- 3. **"컨텍스트/WEB-INF/lib/"** 안에 붙여넣기 한다.

- 이클립스 Dynamic Web Project에서는
- **"컨텍스트/WebContent/WEB-INF/lib/"**안에 붙여넣고 프로젝트를 refresh(새로고침)해주자