

web-server

web-server = web-client의 요청을 받아서 처리 후 응답하는 컴퓨터

tomcat = web-server + web-application-server

web-server = http 요청 + html5 + http 응답

web-application-server = servlet-container + jsp-container

html + servlet + jsp

11, 12-jsp

자바클래스에 html태그 포함 형식 *.java	jsp-html파일내부 자바문장 포함 형식 *.jsp -> *.java-> *.class(tomcat서버)
class A extends HttpServlet doGet(HttpServletRequest, HttpServletResponse) doPost html태그를 출력문장 매개변수 취급	<html> <% 자바문장 ; %> jspService() 내부 문장이나 지역변수 선언

html	<html> <style> <script> 사용 html파일내용==>jsp포함가능
자바 1> <% %> 2> jsp (요청-응답 필요한 객체 미리 생성-제공) 내장객체 request.getParameter() String str = "출력"; <% out.println(str) ; %> ===> <%= str %>	<%! 멤버변수나 메소드 %>-> 선언문-태그 <%-----%>-> jsp 주석문태그 <!------> <% // %> <%= 변수명 %>-> 출력문-태그 <input type="" value=<%= 변수명 %> <%@ page--> page-지시자태그 -->tomcat 에게 jsp-서블릿 소스 변환시에 알려줌

—내장객체

요청-응답관련 객체	request response out	javax.servlet.HttpServletRequest javax.servlet.HttpServletResponse javax.servlet.jsp.JspWriter(PrintWriter 하위클래스)
오류처리관련객	exception	java.lang.Exception

서식 있음: 글꼴 색: 연한 파랑

채		
---	--	--

web server =web client의 요청을 받아서 처리 후 응답하는 컴퓨터

tomcat = web server + web application server

web server = http 요청 + html5 + http 응답

web application server = servlet container + jsp container

html + servlet + jsp

12장 스크립트태그 + 내장객체

13장 액션태그

<% 자바문장 ; %> 스크립트태그

<jsp:xxxx >자식태그 </jsp:xxx>

-서블릿/jsp

HttpSession session = request.getSession()-->세션객체생성

doGet(HttpServletRequest request , response)-->요청객체생성/응답객체생성

ServletContext application =getServletContext();--> 같은 컨텍스트내부 모든 파일 정보 공유

try{.....}catch(Exception exception) {}--> 예외객체생성

- jsp소스-->서블릿소스 변환--> _jspService() " 지역변수 "

<%! void m(....) {내장객체사용x} %>

동작 – 포함/이동/파라미터추가/자바객체생성동작 태그

"new" MemerDTO()

<% request.setAttribute("c", "배열/객체/ 문자열/숫자/논리값.); %>

<jsp:include page="동일서버의동일컨텍스트/포함파일명(html,jsp,서블릿url)">

<jsp:param name="a" value="문자열/숫자/논리값" />

</jsp:include>

<% 응답 포함x (무시), 응답 내용 만들어서 처리 과정 ...%>

<% request.setAttribute("c", "배열/객체/ 문자열/숫자/논리값.); %>

<jsp:forward page="동일서버의동일컨텍스트/이동파일명(html,jsp,서블릿url)">

<jsp:param name="a" value="문자열/숫자/논리값" />

</jsp:include>

- 액션태그 (자바객체생성동작)

<u><jsp:useBean /></u>	<u><% test.A a1 = new test.A();</u> <u>request.setAttribute("a1", a1);</u> <u>%></u> <u><jsp:useBean id="a1" class="test.A" /></u> <u>-->"기본생성자(매개변수 없는 생성자) 호출 변환"</u> <u>bean/ beans ---> 자바객체들</u>
<u><jsp:setProperty /></u>	<u>setter메소드호출동작태그</u> <u><% test.A a1 = new test.A();</u> <u>a1.setName("이자바");</u> <u>%></u> <u><jsp:useBean id="a1" class="test.A" /></u> <u><jsp:setProperty name="a1" property="name" value="이</u> <u>자바" /></u>
<u><jsp:getProperty /></u>	<u>getter메소드호출동작태그</u> <u><% test.A a1 = new test.A();</u> <u>a1.setName("이자바");</u> <u>%></u> <u><%=a1.getName()%></u> <u><jsp:useBean id="a1" class="test.A" /></u> <u><jsp:setProperty name="a1" property="name" value="이</u> <u>자바" /></u> <u><jsp:getProperty name="a1" property="name" /></u>

서식 있는 표

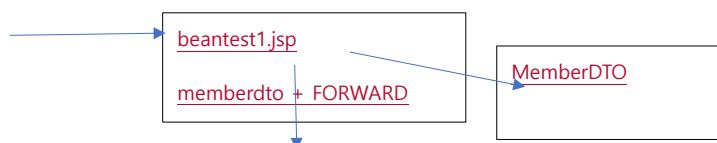
서식 있음: 글꼴: 굵게 없음

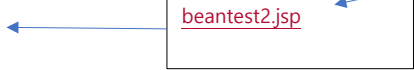
서식 있음: 글꼴: 굵게

서식 있음: 글꼴: 굵게 없음

MemberDTO 클래스

6개 변수 = SETTER/GETTER





<jsp:useBean

id	객체변수명
class	객체생성타입 패키지명.클래스명
scope	page="기본값" - 다른 jsp 과 공유x request - forward/include 파일 공유 session - 같은 브라우저 파일 공유 application - 모든 브라우저 파일 공유

로그인했고 메뉴 내정보수정 링크 클릭했다

updateform.jsp?id=bean&pw=1111

수정정보입력폼
아이디: bean 수정불가
암호: 1111 수정불가
이름, 폰, 이메일 수정 가능
가입일 : 오늘날짜 수정불가
수정 버튼 -
updateprocess.jsp

updateprocess.jsp
1>6개폼 입력값 가져오기
2>MemberDTO 객체로 생성하기
<useBean setProperty, getProperty
3> MemberDAO 클래스 updateMember() 호출하여 리턴 MemberDTO 객체
4> updatesuccess.jsp 이동하면서 전달

updatesuccess.jsp
1> 전달받은 객체 응답

-MemberDAO 클래스 추가구현
MemberDTO updateMember(MeberDTO dto)
메소드 정의
1>update member
set name=?, email=?, phone=?
where id=? and pw=?
2> int 리턴 >= 1 이면
3> SELECT * FOM MEMBER WHERE ID=?
AND PW=?
4> 조회결과를 MemberDTO 객체로 저장

서식 있는 표
서식 있음: 글꼴 색: 빨강

서식 있음: 글꼴 색: 빨강

html 파일 == HTML + CSS + JAVA SCRIPT 복합

<% %>

내장객체

<jsp: --> 6가지>

14장 표현언어와 jstl

EXPRESSION LANGUAGE== EL

<SCRIPT>

\$(document).ready(..)==>jquery

\$(el 변수 el 연산자)

\$(100) --> 100

\$(백+100) --> 공백

<% out.println(); %>

value='<%=a %>'

value=null

변수

논리값/정수/실수/문자열/null

\${true}

\${false}

\${3.14}

\${'aaaaa'}

\${"aaaaa"}

\${null}

연산자

+ 산술연산자

+= 문자열결합연산자

\$('백' + 1) ---> error

\$('백' += 1)

$\$(null + 1) ==> null$ '+' 연산시 자동 0
 $\$(10 / 3) --> 실수$
 $\$(10 \% 3) --> 나머지$
 $\$(empty 1) --> ?$

- el과 jsp태그

pageContext 내장객체

- jsp 파일당 1개씩

- 같은 jsp 내부 el값 공유

pageContext 내장객체

<% int i = 10;

pageContext.setAttribute("jsp_i" , i);

%>

pageContext 내장객체

$\${jsp_i}$ $\${j = 100; } ==> 자동$

pageContext.setAttribute("j" , j);

<% int i = 요청파라미터값정수변환

<% pageContext.getAttribute("j")%>

서식 있음: 글꼴 색: 빨강

| jsp태그 | el | |
|--|---|------------------|
| pageContext.setAttribute("a", "b");
pageContext.getAttribute("a"); | $\${a}$ | 같은 jsp파일 내부 |
| request.setAttribute("a", "b");
request.getAttribute("a"); | $\${a}$ | 다른 jsp파일 (이동/포함) |
| session.setAttribute("a", "b");
session.getAttribute("a"); | $\${a}$ | 다른jsp파일(동일브라우저) |
| application.setAttribute("a", "b");
apliation.getAttribute("a"); | $\${a}$ | 서버지속-다른브라우저 |
| pageContext.setAttribute("a", "b1");
request.setAttribute("a", "b2"); | $\${a} --> b1$
$\${requestScope.a} --> b2$
$\${sessionScope.a} --> "$
$\${applicationScope.a} -->$ | |

서식 있는 표

| | | |
|--|--|--|
| | | |
|--|--|--|