Page 디렉티브 태그 기능 사용법

Page 디렉티브 태그는 현재 jsp 페이지에 대한 정보를 설정하는 태그

Jsp 페이지가 생성할 콘텐츠 유형의 문서, 사용할 자바 클래스, 오류 페이지 설정, 세션 사용 여부 등 jsp 컨테이너가 jsp 페이지를 실행하는데 필요한 정보를 설정할 수 있음.

최상단에 선언하는 걸 권장함.

속성

1.Language: 사용할 프로그래밍 언어 설정

2.contentType: 현재 jsp 페이지 콘텐츠 유형을 설정하는데 사용.

Text/html, text/xml, text/plain 등이 있음.

기본 값은 txt/html로 html 출력하는 jsp페이지는 contentType을 설정할 필요가 없습니다.

ContentType속성은 문자열 세트(charset)을 설정하여 jsp 페이지 전체 문자열을 변경할 수 있음.

3.pageEncoding속성

jsp페이지 문자 인코딩 유형을 설정하는데 사용함

문자 인코딩 유형의 기본값은 ISO-8859-1

4.import 속성

현재 jsp 페이지에서 사용할 자바 클래스를 설정하는데 사용함.

둘 이상의 자바클래스를 포함하는 경우 쉼표로 구분하여 연속해서 여러 개의 자바 클래스 설정 가능함.

<%@page import=”java.io.\*”%>

[%@page import=”java.lang.\*”%](mailto:%25@page%20import=)

5.session속성

현재 jsp페이지의 http 세션 사용 여부를 설정하는 데 사용함

기본 값은 세션을 자동으로 사용하는 true. 사용하지 않으려면 false.

<%@page session = “true”%>

6.buffer

jsp페이지의 출력 버퍼 크기를 설정하는데 사용함,

none과 버퍼크기로 설정 가능함.

<page buffer = “none”%>

버퍼크기 32kb설정?: <page buffer = “32kb”%>

7.autoflush

자동으로 출력 버퍼를 비우는 것

출력 버퍼가 채워진 경우 자동으로 버퍼에 있는 데이터를 웹브라우저로 보냄

버퍼를 비울지 버퍼 오버플로의 예외를 나타낼지 등의 처리를 함.

그렇다면 버퍼는 무엇일까?: (내일 찾아두기)

<%@page autoFlush = “true”%>

8.isThreadSafe

jsp페이지에서 멀티스레드의 동작을 제어하는 데 사용함

여러 요청이 동시에 수신되는 경우 jsp 페이지가 어떻게 동작해야 하는지 jsp컨테이너에 지시하여 응답할 수 있게 함.

기본 값은true

False 하면 jsp페이지에 대한 요청이 순차적으로 처리됨

<%@ page isThreadSafe=”true”%>

9.info 설정

현재 jsp페이지에 대한 설명을 설정하는데 사용함.

<%@page info = “Date 클래스 이용한 날짜 출력하기%>//웹페이지 설명 설정하기 위해 page 디렉티브 태그의 info속성을 작성함.

Today is <%=new java.util.Date()%>// 현재 날짜 출력하도록 표현문 태그 사용.

10.errorPage

오류가 발생하면 특정 오류 페이지로 이동하는데 사용함

페이지 두 개로 나눠서 실행<%@page errorpage = “page\_errorpage.jsp”%> //이동할 오류 페이지 jsp설정

<%

String str = null; // 변수 str에 값을 저장하고 저장된 값을 출력하도록 스크립트태그 작성

Out.println(str.toString());

%>

11.isErrorPage

jsp페이지가 오류페이지인지 여부를 설정하는 데 사용함.

기본 값은 false. 예외 처리를 위한 내장 객체인 exception 변수를 사용할 수 없음.

속성값을 true로 설정하면 현재 페이지 오류.

그래서 결론은 뭐? 내장객체인 exception을 따로 기재해줘야 한다

<%@page isErrorPage=”true”%> // 현재 jsp페이지가 오류 페이지인지 여부를 설정하려고 속성 작성함

12.isELIgnored

jsp페이지의 표현 언어(expression language) 사용 여부를 설정하는데 사용함. 기본값은 false.

Jsp 페이지 표현 언어의 표현식 ${}을 사용할 수 있다.

13.isScriptingEnabled

속성은 스크립트 태그(선언문, 스크립틀릿,표현문) 사용 여부를 설정하는데 사용함.

<< 액션태그>>

1. 액션태그의 개요

서버나 클라이언트에게 “어떤 행동을 하도록 명령하는 태그”

주석, 디렉티브 태그와 함께 jsp 페이지 구성하는 태그임.

페이지와 페이지 사이를 제어하거나, 다른 페이지의 실행 결과 내용을 현재 페이지에 포함하거나, 자바빈즈 등의 다양한 기능 제공.

액션태그는 xml형식인 <jsp: …/>를 사용함.

1. forward 액션 태그 기능과 사용법

“현재 jsp 페이지에서 다른페이지로 이동하는 태그”

그 전까지 출력 버퍼에 저장되어 있던 내용을 모두 삭제함.

<jsp:forward page=”파일명”/>

또는

<jsp:forward page=”파일명”></jsp:forward>

페이지 속성값: 현재 페이지에서 이동할 페이지의 외부 파일명

외부파일은 현재 jsp페이지와 같은 디렉에 있으면 파일명만 설정

아닌 경우 url 설정해야 함.

1. include 액션 태그 기능과 사용법  
   <jsp:include page=”파일명” flush=”false”/>

기본값이 false. true로 설정하면 외부 파일로 제어가 이동할 때 현재 페이지가 지금까지 출력 버퍼에 ㄷ저장된 내용을 웹 브라우저에 출력하고 출력 버퍼를 비움.

1. param 액션 태그 기능과 사용법

현재 jsp페이지에서 다른 페이지에 정보를 전달하는 태그

단독으로 사용하지 못함!! 그래서 <jsp:forward>나 <jsp:include> 태그의 내부에 사용됨

<jsp:forward page=”파일명”>

<jsp:param name =”매개변수명1” value=”매개변수값1”/>

<jsp:param name = “매개변수명1” value=”매개변수값2”/>

</jsp:forward>

<**body**>

<**p**> 아이디: <%=request.getParameter("id") %>

<%

String name = request.getParameter("name");%>

<**p**> 이름: <%=java.net.URLDecoder.decode(name) %>

</**body**>

여기 있는 내용을

<**h3**>param 액션 태그</**h3**>

<**jsp:forward** page = *"param01\_data.jsp"*>

<**jsp:param** name=*"id"* value=*"admin"*/>

<**jsp:param** name=*"name"* value=*'*<%=java.net.URLEncoder.encode("관리자")%>*'* />

</**jsp:forward**>

<**p**>JAVA Server Page

여기로 가져감

1. 자바빈즈 액션 태그의 기능과 사용법

동적 컨텐츠 개발을 위해 자바 코드를 사용하여 자바 클래스로 로직을 작성하는 방법

1. 웹브라우저가 jsp 페이지에 요청
2. jsp페이지는 자바빈즈와 통신
3. 자바빈즈가 데이터베이스에 연결
4. jsp페이지가 브라우저에 응답
5. package com.dto;
6. import java.io.Serializable;
7. import javax.tools.JavaCompiler;
8. public class MEmberBean implements java.io.Serializable{
9. private int id;
10. private String name; //멤버 변수(프로퍼티)정의
11. public MEmberBean() {
12. }//기본 생성자
13. public int getId() {
14. return id;
15. }
16. public void setId(int id) {
17. this.id = id;
18. }
19. public String getName() {
20. return name;
21. }
22. public void setName(String name) { //모든 멤버 변수 프로퍼티는 게터 세터 메소드가 있어야 함.
23. this.name = name; //게터는 저장된 값을 가져올 수 있는 메소드. 세터는 멤버 변수게 값을 저장할 수 있음!
24. }
25. }

useBeran액션 태그로 자바빈즈 사용하기

|  |
| --- |
| <jsp:useBean id=”자바빈즈 식별이름” class=”자바빈즈 이름” scope=”범위”/>  -><jsp:useBean id=”member” class=”com.dto.MemberBean” scope=”page”/>  <%@page contentType=”text/html; charset = utf-8”%>  <html>  <head>  <title>액션 태그</title>  <body>  <jsp:useBean id=”member” class=”com.dto.MemberBean” scope=”requeset”/>  <p> 아이디: <%=person.getId()%>  <p> 이름 : <%=person.getId()%>  <%  person.setId(20182005);  person.setName(“홍길동”;  </body>  </head>  </html> |

5-2)useBean 액션태그로 자바빈즈 사용하기

.jsp페이지에서 자바빈즈를 사용하기 위해 **실제 자바클래스를 선언하고** **초기화**하는 태그

객체 발견되지 않으면 빈 객체를 생성함

[useBean 액션태그 사용 예시]

|  |
| --- |
| <jsp:useBean **id**=”member” **class**=”com.dto.MemberBean” **scope**=”page”/>  **Id**는 자바빈즈 식별하기 위한 이름  **Class** 패키지이름 포함 자바빈즈 이름  **Scope** 자바빈즈 저장되는 영역 |

예제

1. 자바빈즈 person클래스 먼저 생성, 게터세터 생성. 액션태그에 person클래스를 사용해서 아이디와 이름 출력

package ch04.com.dto;

public class Person {

private int id = 20181004;

private String name = "홍길순";

public Person() {

// **TODO** Auto-generated constructor stub

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

}

2.자바빈즈 person으로 아이디와 이름 설정해서 출력하는 jsp파일 useBean 생성

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**jsp:useBean** id=*"person"* class=*"ch04.com.dto.Person"* scope = *"request"* />

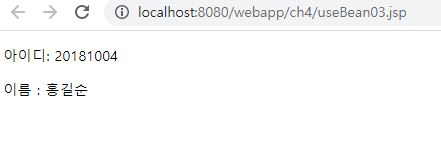
<**p**> 아이디: <%=person.getId() %>

<**p**> 이름 : <%=person.getName() %>

</**body**>

</**html**>

결과:



<값 변경하기>

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**jsp:useBean** id=*"person"* class=*"ch04.com.dto.Person"* scope=*"request"* />

<**p**> 아이디: <%=person.getId() %>

<**p**> 이름 : <%=person.getName() %>

<% person.setId(20182005); //set으로 기존값 변경

person.setName("홍길동"); //set으로 기존값 변경

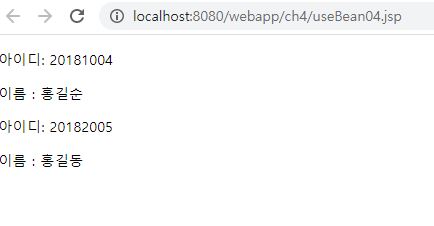
%>

<**jsp:include** page=*"useBean03.jsp"*/>

</**body**>

</**html**>

결과



5-3)setProperty **액션태그**로 프로퍼티 값 저장하기

useBean액션 태그와 함께 자바빈즈의 setter() 메소드에 접근해서 자바빈즈의 멤버 변수인 **프로퍼티 값을 저장**!

필드를 프로퍼티라고 함

|  |
| --- |
| <jsp:setProperty name=”자바빈즈 식별이름” property=”프로퍼티 이름” value=”값”/>  Name: id속성 값으로 설정된 자바빈즈 식별 이름  Property: 자바빈즈 프로퍼티 이름. \* 사용하면 모든 요청 파라미터가 자바빈즈 프로퍼티 setter()메소드에 전달됨  Value:변경할 자바빈즈 프로퍼티 값  Param:프로퍼티 **값을 전달**하는 요청 파라미터의 이름. **Param value 동시사용 불가능** |

<예제>

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**jsp:useBean** id=*"person"* class=*"ch04.com.dto.Person"* scope=*"request"* />

<**jsp:setProperty** name=*"person"* property=*"id"* value=*"20182005"* />

<**jsp:setProperty** name=*"person"* property=*"name"* value=*"홍길동"* />

<**p**> 아이디: <%out.println(person.getId());%>

<**p**>이름: <% out.println(person.getName()); %>

</**body**>

</**html**>

5-4)getProperty

getter메소드에 접근하여 자바빈즈 멤버 변수인 프로퍼티 값을 가져오는 태그

|  |
| --- |
| 액션태그 사용 예시:  <jsp:getProperty name”자바빈즈 식별이름” property=”프로퍼티 이름” />  자바빈즈 프로퍼티(스크립틀릿) 값 출력 예시:  <%out.println(member.getName());%> |

예시)

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**jsp:useBean** id=*"person"* class=*"ch04.com.dto.Person"* scope=*"request"* />

//정보 가져올 경로 클래스 설정

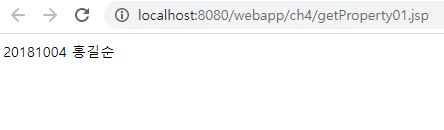
<**jsp:getProperty** name=*"person"* property=*"id"* /> //getProperty 액션태그

<**jsp:getProperty** name=*"person"* property=*"name"* />//getProperty 액션태그

</**body**>

</**html**>

결과)



예제2)

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**jsp:useBean** id=*"person"* class=*"ch04.com.dto.Person"* scope=*"request"* />

<**jsp:setProperty** name=*"person"* property=*"id"* value=*"20182005"* />

<**jsp:setProperty** name=*"person"* property=*"name"* value=*"홍길동"* />

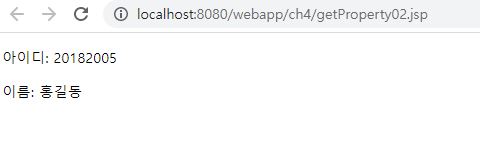
<**p**> 아이디: <**jsp:getProperty** property = *"id"* name=*"person"* />

<**p**> 이름: <**jsp:getProperty** property = *"name"* name=*"person"* />

</**body**>

</**html**>

결과)



**<<05.내장 객체>**

1. **내장객체 개요**
2. **Request 내장 객체의 기능과 사용법**
3. **Response 내장객체 기능과 사용법**
4. **Out 내장 객체 기능과 사용법**
5. **내장객체 개요**
6. **request 내장 객체 기능과 사용법**

\*독립적인 값을 가져올지 멀티값을 가져올지에 따라 사용되는 함수가 구분됨!\*

getParameter(): 요청 파라미터 이름이 name인 값을 전달받음. 값 없으면 null을 반환.

parameterValues(): 모든 요청 파라미터 이름이 name인 값을 배열 형태로 전달받음(값이 두 개 이상)

getParameterNames(): value와 name값을 동시에 알아보고 싶을 때

getParameterMap():key와 value구조로도 전부 얻어볼 수 있음

2-1)요청 파라미터 값 출력하기 getParameter 예제

<%@ **page** contentType=*"text/html; charset=EUC-KR"*%>

<**html**>

<**head**>

<**title**>Implicit Objects</**title**>

</**head**>

<**body**>

<**form** action = *"request01\_process.jsp"* method = *"post"*>

<**p**>아이디 : <**input** type = *"text"* name = *"id"*>

<**p**>비밀번호: <**input** type=*"text"* name = *"passwd"*>

<**p**> <**input** type=*"submit"* value=*"전송"* /> submit:**버튼이 자동으로 만들어지고 이벤트까지 걸림**

</**form**>

</**body**>

</**html**>

스크립틀릿

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

<**html**>

<**head**>

<**title**>Implicit Objects</**title**>

</**head**>

<**body**>

<%

request.setCharacterEncoding("utf-8");

String userid = request.getParameter("id");

String password = request.getParameter("passwd");

%>

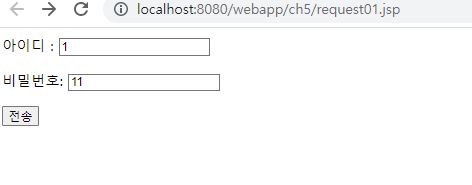
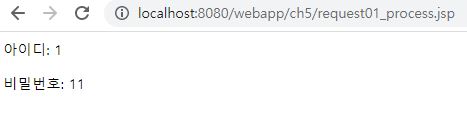
<**p**> 아이디: <%=userid %>

<**p**> 비밀번호: <%=password %>

</**body**>

</**html**>

결과)

전송!

1. **response 내장 객체 기능과 사용법**

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Implicit Objects</**title**>

</**head**>

<**body**>

<**form** action = *"response01\_process.jsp"* method=*"post"*>

<**p**> 아이디: <**input** type=*"text"* name=*"id"*>

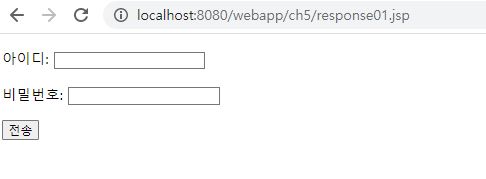
<**p**> 비밀번호: <**input** type=*"text"* name=*"passwd"*>

<**p**> <**input** type=*"submit"* value =*"전송"*>

</**form**>

</**body**>

</**html**>



로그인 실패/성공

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Implicit Objects</**title**>

</**head**>

<**body**>

<**p**>로그인을 실패했습니다.

<**p**><**a** href=*"./response01.jsp"*> 로그인가기</**a**> //로그인페이지로 돌아가게 링크 걸어둠

</**body**>

</**html**>

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Implicit Objects</**title**>

</**head**>

<**body**>

로그인을 성공했습니다

</**body**>

</**html**>

로그인할 때 넘어온 값에 대한 if else문

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<%

request.setCharacterEncoding("utf-8");

String userid = request.getParameter("id");

String password = request.getParameter("passwd");

if(userid.equals("관리자")&&password.equals("1234")){

response.sendRedirect("response01\_success.jsp"); //sendRedirect 해당페이지로 보내겠다

}else{

response.sendRedirect("response01\_failed.jsp");

}

%>

</**body**>

</**html**>

**Chapter 6. 폼 태그**

1.폼 처리의 개요

**2.Form 태그 기능과 사용법**

3.Input 태그의 기능과 사용법

4.Select 태그의 기능과 사용법

5.Textarea 태그의 기능과 사용법

6.폼데이터 처리하기

**2.Form 태그란?**

Html 문법 사용해요~

|  |
| --- |
| <form 속성1=”값1” [속성2=”값2”…]>  //다양한 입력 양식 태그<input><select><textarea>  </form> |

get/post 차이 꼭 인지하고 갑시다!

<https://velog.io/@songyouhyun/Get%EA%B3%BC-Post%EC%9D%98-%EC%B0%A8%EC%9D%B4%EB%A5%BC-%EC%95%84%EC%8B%9C%EB%82%98%EC%9A%94>

2-1)input

|  |
| --- |
| <input 속성1=”값1” [속성2=”값2”…]> |

일단 여기까지만 진도 나갔고

다시 돌아갑시다~

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!DOCTY**PE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**form** action=*"request02\_process.jsp"* method=*"post"*>

취미: <**br**>

독서: <**input** type=*"checkbox"* name=*"hobby"*>

운동: <**input** type=*"checkbox"* name=*"hobby"*>

영화: <**input** type=*"checkbox"* name=*"hobby"*>

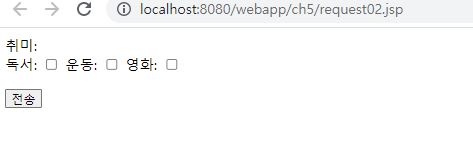
<**p**> <**input** type=*"submit"* value=*"전송"* />

</**form**>

</**body**>

</**html**>

결과)



<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Implicit Objects</**title**>

</**head**>

<**body**>

<%

request.setCharacterEncoding("utf-8");

String[] hobby = request.getParameterValues("hobby");

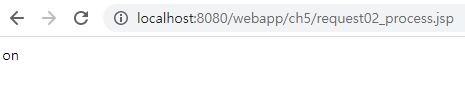
out.println(hobby[0]);

%>

</**body**>

</**html**>

위에 jsp가 있으면체크박스 체크후



이 결과창이 뜸

Html로 이름,이메일,자기소개, 이름확인버튼 테이블 만들기

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**form** action=*"#"*>

<**table** border=*1*>

<**tr**>

<**th** align=*"center"*>이름</**th**> // align: 정렬

<**td** align=*"center"*><**input** type=*"text"* name=*"myname"* size=*"50"*>

</**td**>

</**tr**>

<**tr**>

<**th** align=*"center"*>E-mail</**th**>

<**td** align=*"center"*><**input** type=*"text"* name=*"email"* size=*"50"*>

</**td**>

</**tr**>

<**tr**>

<**th** align=*"center"*>자기소개</**th**>

<**td** align=*"center"*><**textarea** name=*"content"* rows=*"14"* cols=*"50"*></**textarea**> //>전체가 여러 문장을 저장할 수 있는 컴포넌트임

</**td**>

</**tr**>

<**tr**>

<**td** colspan=*"2"* align=*"center"*>

<**input** type=*"submit"* value=*"이름확인"*>

</**td**>

</**tr**>

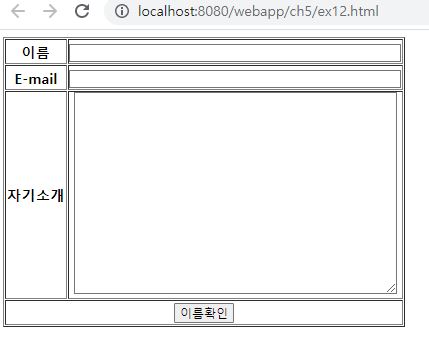
</**table**>

</**form**>

</**body**>

</**html**>

결과물)



Html로 라디오, 체크박스 연습

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**h3**>\*라디오, 체크박스 연습\*</**h3**>

<**HR**>

<**BR**>

<**form** action=*"#"*>

좋아하는 색깔 <**BR**>

<**input** type=*"radio"* name=*"color"* value=*"빨간색"* checked>빨간색<**BR**> //마지막에 있는 checked는 미리 체크를 해주는 표시임

<**input** type=*"radio"* name=*"color"* value=*"파란색"*>파란색<**BR**>

<**input** type=*"radio"* name=*"color"* value=*"노란색"*>노란색<**BR**>

좋아하는 과일<**br**>

<**input** type=*"checkbox"* name=*"fruit"* value=*"사과"*>사과<**br**>

<**input** type=*"checkbox"* name=*"fruit"* value=*"오렌지"*>오렌지<**br**>

<**input** type=*"checkbox"* name=*"fruit"* value=*"바나나"*>바나나<**br**>

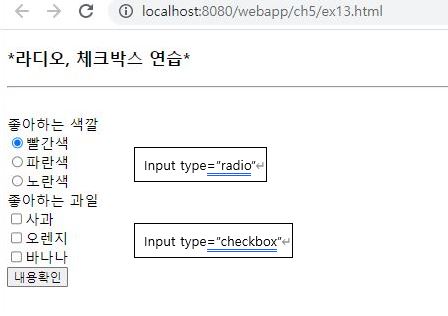
<**input** type=*"submit"* value=*"내용확인"*><**br**>// value는 버튼 이름설정 가능

</**form**>

</**body**>

</**html**>

결과



\*radio는 중복 체크 안 됨. Checkbox는 중복 체크 가능함!!

Html 리스트, 리스트 박스 연습

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**h3**>\*리스트, 리스트 박스 연습\*</**h3**><**hr**><**br**>

<**form** action =*"#"*>

좋아하는 색깔<**br**>

<**select** name=*"color"* size = *"1"*>

<**option** value=*"빨간색"*>빨간색 //리스트 아이템

<**option** value=*"파란색"*>파란색

<**option** value=*"노란색"*>노란색

</**select**><**P**>

좋아하는 과일<**br**>

<**select** name=*"fruit"* size=*"3"* multiple> // multiple 멀티체킹 가능

<**option** value=*"사과"*>사과

<**option** value=*"배"*>배

<**option** value=*"오렌지"*>오렌지

</**select**>

<**input** type = *submit* value=*"내용확인"*><**br**>

</**form**>

</**body**>

</**html**>

결과



Html input 요소 연습

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**h1**>input 요소</**h1**>

<**form** action=*"#"*>

<**fieldset**> // 테두리 만들 수 있음

<**legend**>회원가입</**legend**> // <fieldset> 요소의 캡션(caption)을 정의할 때 사용합니다.

<**p**>아이디 : <**input** type = *"text"* size=*"12"* maxlength=*"8"*/><**p**>

// maxlength:글자수 제한

<**p**>아이디 : <**input** type = *"password"* size=*"12"* maxlength=*"8"*/><**p**>

<**p**> 메일수신여부:

<**input** type = *"radio"* size=*"y"* name=*"receive"*/>예

<**input** type = *"radio"* size=*"n"* name=*"receive"*/>아니요

</**p**>

<**p**>관심 분야:

<**input** type = *"checkbox"* value=*"HTML"* name=*"chk1"*/>HTML

<**input** type = *"checkbox"* value=*"CSS"* name=*"chk2"*/>CSS

<**input** type = *"checkbox"* value=*"Javascript"* name=*"chk3"*/>Javascript

</**p**>

<**p**>

<**input** type=*"submit"* value=*"전송"*/>

<**input** type=*"reset"* value=*"취소"*/> //reset: 값 입력한 걸 초기화 시켜줌(다 지워주는 이벤트가 걸려있음. 내장함수 이용한 거임)

<**input** type=*"button"* value=*"확인"*/> //이건 일반적인 버튼임. 이벤트 걸려있지 않은 일반 버튼임. 그럼 이런 버튼은 이벤트 걸 수 있는 방법 없나요? 나중에 배움

<**input** type=*"image"* src=*"img/다운로드.jpg"* alt =*"검색"* /> //이미지 사용. 이미지 찾을 수 없을 때 대체해서 텍스트 나오게끔 하는게 alt.

</**p**>

<**p**>파일 올리기: <**input** type=*"file"* /></**p**> type속성의 값으로 file을 주게 됨. 파일 여는 기능이 내장되어있음! 파일 업로드 시간에 다시 다룰 예정임.

<**p**><**input** type =*"hidden"* /></**p**> // elements가 표현되지 않음. Ex.사용자가 몇 번째 회원인지.. 회원 개인 정보 출력할 때 굳이 개인에게 보여줄 필요가 없는 것들을 가리고 싶을 때 사용함.(하지만 시스템엔 필요한 것들이기에.. 걍 숨기는 거임.)

이와 비슷한 걸로Readonly,disable 있음. 보여지긴 하지만 비활성화 되는 것.

Id,pw 등 보이는 회원정보수정창에서 수정하지 말아야 할 정보들이 이짜나용.. 그걸 비활성화 할 때 readonly와 disable을 사용합니당…

하지만 값을 DB에 넘길 때는 readonly를 써야하고 넘기지 않아도 될 땐 disable을 사용합니당…

</**fieldset**>

</**form**></**body**>

</**html**>

결과)



회원가입창 Jsp

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**form** action=*"form04\_process.jsp"* name=*"member"* method=*"post"*>

<**p**> 아이디: <**input** type=*"text"* name=*"id"*> <**input** type=*"button"*

value=*"아이디 중복검사"*>

<**p**> 비밀번호 : <**input** type = *"password"* name=*"passwd"*>

<**p**> 이름: <**input** type=*"text"* name=*"name"*>

<**p**>연락처 : <**select** name=*"phone1"*>

<**option** value=*"010"*>010</**option**>

<**option** value=*"011"*>011</**option**>

<**option** value=*"016"*>016</**option**>

<**option** value=*"017"*>017</**option**>

<**option** value=*"019"*>019</**option**>

</**select**> - <**input** type = *"text"* maxlength=*"4"* size=*"4"* name=*"phone2"*> -

<**input** type=*"text"* maxlength=*"4"* size=*"4"* name=*"phone3"*>

<**p**>성별: <**input** type = *"radio"* name=*"sex"* value=*"남성"* checked> 남성

<**input** type = *"radio"* name=*"sex"* value=*"여성"*> 여성

<**p**>취미: 독서<**input** type=*"checkbox"* name=*"hobby1"* checked>

운동<**input** type=*"checkbox"* name=*"hobby2"* >

영화<**input** type=*"checkbox"* name=*"hobby3"* >

<**p**><**textarea** name=*"comment"* cols=*"30"* rows=*"3"*

placeholder=*"가입인사를 입력해주세요"*></**textarea**>

<**p**><**input** type=*"submit"* value=*"가입하기"*>

<**input** type=*"reset"* value=*"다시쓰기"*>

</**form**>

</**body**>

</**html**>

결과)



<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<% //클라이언트로부터 넘어온 데이터를 받아내는 식

request.setCharacterEncoding("UTF-8");

String id=request.getParameter("id");

String passwd = request.getParameter("passwd");

String name = request.getParameter("name");

String phone1= request.getParameter("phone1");

String phone2 = request.getParameter("phone2");

String phone3 = request.getParameter("phone3");

String sex = request.getParameter("sex");

String hobby1 = request.getParameter("hobby1");

String hobby2 = request.getParameter("hobby2");

String hobby3 = request.getParameter("hobby3");

String comment = request.getParameter("comment");

%>

<**p**> 아이디: <%=id %>

<**p**> 비밀번호 : <%=passwd%>

<**p**> 이름: <%=name%>

<**p**>연락처 : <%=phone1%>-<%=phone2%>-<%=phone3%>

<**p**> 성별: <%=sex%>

<**p**> 취미 : <%=hobby1%><%=hobby2%><%=hobby3%>

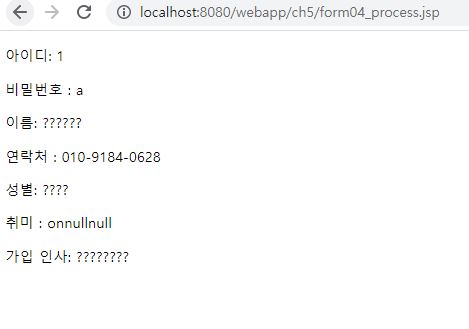
<**p**> 가입 인사: <%=comment%>

</**body**>

</**html**>

결과를 출력하게 해 줌

결과)



그렇다면 hobby 1,2,3 으로 나뉜게 아니라 hobby로 통으로 묶인다면?

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**form** action=*"form05\_process.jsp"* name=*"member"* method=*"post"*>

<**p**> 아이디: <**input** type=*"text"* name=*"id"*> <**input** type=*"button"* value=*"아이디 중복검사"*>

<**p**> 비밀번호 : <**input** type = *"password"* name=*"passwd"*>

<**p**> 이름: <**input** type=*"text"* name=*"name"*>

<**p**>연락처 : <**select** name=*"phone1"*>

<**option** value=*"010"*>010</**option**>

<**option** value=*"011"*>011</**option**>

<**option** value=*"016"*>016</**option**>

<**option** value=*"017"*>017</**option**>

<**option** value=*"019"*>019</**option**>

</**select**>

- <**input** type = *"text"* maxlength=*"4"* size=*"4"* name=*"phone2"*> -

<**input** type=*"text"* maxlength=*"4"* size=*"4"* name=*"phone3"*>

<**p**>성별: <**input** type = *"radio"* name=*"sex"* value=*"남성"* checked> 남성

<**input** type = *"radio"* name=*"sex"* value=*"여성"*> 여성

<**p**>취미: 독서<**input** type=*"checkbox"* name=*"hobby"* checked>

운동<**input** type=*"checkbox"* name=*"hobby"* >

영화<**input** type=*"checkbox"* name=*"hobby"* >

<**p**><**textarea** name=*"comment"* cols=*"30"* rows=*"3"*

placeholder=*"가입인사를 입력해주세요"*></**textarea**>

<**p**><**input** type=*"submit"* value=*"가입하기"*>

<**input** type=*"reset"* value=*"다시쓰기"*>

</**form**>

</**body**>

</**html**>

값 저장 jsp

<%@ **page** language=*"java"* contentType=*"text/html; charset=EUC-KR"*

pageEncoding=*"EUC-KR"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"EUC-KR"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<%

request.setCharacterEncoding("UTF-8");

String id = request.getParameter("id");

String passwd = request.getParameter("passwd");

String name = request.getParameter("name");

String phone1= request.getParameter("phone1");

String phone2 = request.getParameter("phone2");

String phone3 = request.getParameter("phone3");

String sex = request.getParameter("sex");

String[] hobby = request.getParameterValues("hobby");

String comment = request.getParameter("comment");

%>

<**p**>아이디: <%= id%>

<**p**> 비밀번호 : <%=passwd%>

<**p**> 이름: <%=name%>

<**p**>연락처 : <%=phone1%>-<%=phone2%>-<%=phone3%>

<**p**> 성별: <%=sex%>

**<p> 취미 : <%**

**if (hobby != null){**

**for (int i=0; i<hobby.length; i++){**

**out.println("" + hobby[i]);**

**}**

**} // 표현식이 아닌 스크립틀릿으로 진행**

%>

<**p**> 가입 인사: <%=comment%>

</**body**>

</**html**>

\*클라이언트에서 null값이 넘어오지 못하게 하는 방법: 유효성검사

하지만 null체크 해주는 건 기본이라서 자바스크립트를 이용해서 한다구 함…. 우왕

파라미터 표현

<%@ **page** language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**form** action=*"form06\_process.jsp"* name=*"member"* method=*"post"*>

<**p**> 아이디: <**input** type=*"text"* name=*"id"*> <**input** type=*"button"* value=*"아이디 중복검사"*>

<**p**> 비밀번호 : <**input** type = *"password"* name=*"passwd"*>

<**p**> 이름: <**input** type=*"text"* name=*"name"*>

<**p**>연락처 : <**select** name=*"phone1"*>

<**option** value=*"010"*>010</**option**>

<**option** value=*"011"*>011</**option**>

<**option** value=*"016"*>016</**option**>

<**option** value=*"017"*>017</**option**>

<**option** value=*"019"*>019</**option**>

</**select**>

- <**input** type = *"text"* maxlength=*"4"* size=*"4"* name=*"phone2"*> -

<**input** type=*"text"* maxlength=*"4"* size=*"4"* name=*"phone3"*>

<**p**>성별: <**input** type = *"radio"* name=*"sex"* value=*"남성"* checked> 남성

<**input** type = *"radio"* name=*"sex"* value=*"여성"*> 여성

<**p**>취미: 독서<**input** type=*"checkbox"* name=*"hobby"* checked>

운동<**input** type=*"checkbox"* name=*"hobby"* >

영화<**input** type=*"checkbox"* name=*"hobby"* >

<**p**><**textarea** name=*"comment"* cols=*"30"* rows=*"3"*

placeholder=*"가입인사를 입력해주세요"*></**textarea**>

<**p**><**input** type=*"submit"* value=*"가입하기"*>

<**input** type=*"reset"* value=*"다시쓰기"*>

</**form**>

</**body**>

</**html**>

이름값들 다수 저장하기 위해 while반복문 실행

<%@**page** import=*"java.util.Enumeration"*%>

<%@ **page** language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!**DOCTYPE** html>

<**html**>

<**head**>

<**meta** charset=*"UTF-8"*>

<**title**>Insert title here</**title**>

</**head**>

<**body**>

<**table** border=*"1"*>

<**tr**>

<**th**>요청 파라미터 이름</**th**>

<**th**>요청 파라미터 값"</**th**>

<%

request.setCharacterEncoding("UTF-8");

Enumeration paramNames = request.getParameterNames();

while (paramNames.hasMoreElements()) { //요소가 있는 만큼 반복할 수 있음

String name = (String) paramNames.nextElement();

out.print("{<tr><td>" + name + "</td>\n");

String paramValue = request.getParameter(name);

out.println("<td>" + paramValue + "</td></tr>\n");

}

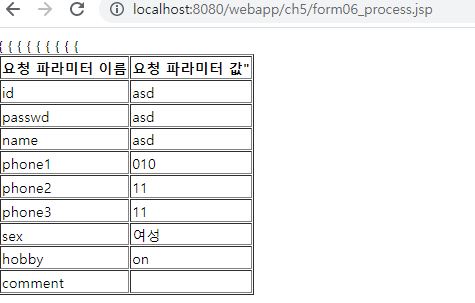
%>

</**table**>

</**body**>

</**html**>

결과



그렇다면 여기서 다시 정리!!

스크립틀릿과 표현식 구분

선언

선언에서 변수, 메서드를 선언한다. 변수, 메서드 선언비 반드시 ; (세미콜론)가 필요하다.

<%! 선언문; %>

\* 선언에서 선언된 변수, 메서드는 처음 요청이 있을 때 한 번만 호출된다. 따라서 계속되는 요구에도 변수의 값은 초기화되지 않고, JSP컨테이너(Tomcat 등)를 다시 시작할 때까지 값이 유지된다.

스크립틀릿

스크립틀릿은 JSP 태그에서는 표현할 수 없는 작업을 Java 코드를 작성하고 자유로운 작업을 수행하는 경우에 사용한다. Java 코드를 위한 각 코드에는 반드시 ; (세미콜론)가 필요하다.

<% 스크립틀릿; %>  
\* 스크립틀릿에서 선언 된 변수는 요청 때마다 호출된다. 따라서 그 요청이 있을 때 마다 변수의 값이 초기화된다.

표현식

식에서는 Java 코드를 작성하고 그 결과를 표시한다.  
따라서 실행 결과를 반환 코드 밖에 기술할 수는 없다.  
실행 결과를 리턴하지 void 메서드나 변수의 선언만 식으로 설명할 수 없다.  
식에서는 ; (세미콜론)가 필요하지 않다.

<%= 표현식 %>  
식으로 작성할 수 있는 코드(O)

|  |  |
| --- | --- |
| **코드** | **설명** |
| <%= abc %> | 변수 abc의 값을 표시한다. |
| <%= getValue() %> | 메소드 getValue의 실행 결과를 표시한다. |

식으로 작성할 수 없는 코드(X)

|  |  |
| --- | --- |
| **코드** | **설명** |
| <%= int abc=0 %> | 변수 abc 선언. |
| <%= setValue() %> | 반환 값이 void 메소드 setValue 실행. |

20230405

오늘까지 배운 인풋타입

Input type = text, button, password, radio, checkbox, submit, reset