

스터디회차	6회차		
그룹명	MaveWeb	날짜	4월24일 (제6회모임)
스터디장소	50주년기념관 517호	시간	15시00분 ~ 17시45분
수강과목	웹프로그래밍	담당교수	최주영
스터디	리더: 한아림		
참석자	그룹원: 손효림, 이경화		
학습주제	중간고사 범위 총복습		
학습문제			
<div><div>- 웹프로그래밍</div><div>- JSP 동작원리와 스크립트 요소</div><div>- 스크립트 요소와 폼태그</div><div>- 폼태그와 JSP제어문</div><div>- JSP 지시자와 액션 태그</div><div>- Forward 액션 태그와 자바빈즈</div></div>			
학습문제해결과정			



<모임 사진>

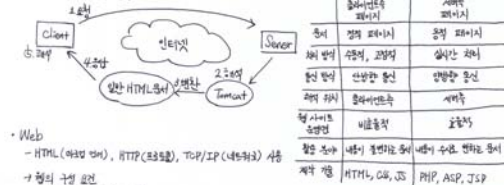
웹 프로그래밍 1주차

<인터 넷>

* 인터넷: 운영체제와 관계없이 모든 기종의 컴퓨터를 TCP/IP로 연결한 통신망

* 인터넷 서비스의 종류: 웹(WWW), 이메일, FTP(파일 전송), Telnet(원격 로그인), DNS(도메인 이름 변환), News

* CS(Client-Server) 환경



* Web

- HTML (하이퍼 텍스트), HTTP (프로토콜), TCP/IP (네트워크) 사용

→ 웹의 구성 요건

- HTTP 프로토콜 (문서 교환)

- HTML (문서 작성)

- URL (문서를 찾아내고 있는 웹상 사이트)

→ 웹 페이지 제작 기술

- HTML → 웹 디자인 (프레이밍, 일러스트 등) → 외부 프로그램 이용 기술 (ASP, PHP, JSP 등 스크립트 언어, DB 기술)

<JSP>

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

* CGI: 동적 페이지를 생성하기 위해 초기에 사용된 기술로, 웹 서버와 연결된 애플릿을 실행한 프로그램 사이 정보를 주고 받는 인터페이스

<내용 정리 1>

- Servlet
 - 산적인 자바 파일의 문제
 - JSP와 달리 자바 코드 안에 HTML 태그를 포함
 - JSP 페이지 → 변환 → 시물릿 → 컴파일 → 실행
 - 내부적으로 처리되어야 할 부분에 보다 유연하게 사용

<JSP 개발 환경>



* 컨테이너: 동적인 페이지 생성을 받아 처리할 프로그램. JSP 페이지에 대한 요청 처리.

2

<내용 정리 2>

웹 프로그래밍 2주차

<Hello JSP!!>

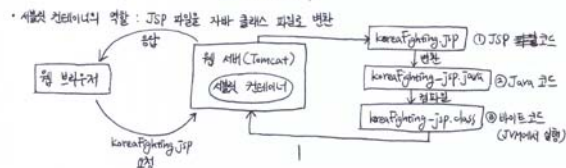
* 이클립스에서 JSP 실행하기

1. Perspective 설정 : Java EE → Web
2. 서버 (Tomcat) 설정
3. Dynamic Web Project 만들기
4. JSP 작성하기
5. JSP 실행하기

<Korea Fighting!!>

- 물리적 공간 (작업하는 공간) : 프로젝트명 / src / main / webapp / 파일명.jsp
- 가상 공간 (작업 결과가 뜨는 공간) : `http://localhost:8080/프로젝트명/파일명.jsp` → 페이지 주소
- 절대적 URL : 프로젝트명 // 소스인 : 프로젝트명 / 폴더 경로 / 파일명
 ex. `file:///C:/homepage/index.html`
`http://www.danone.net/index.html`
`mailto:jychei@swu.ac.kr`
- 상대적 URL
 - `./image/bak.jpg` : 현재 위치
 - `../image/bak.jpg` : 상위 폴더
 - `image/bak.jpg` : 현재 위치
- `` 태그
``
- 하이퍼링크 태그
` 이곳을 클릭하세요 `

<JSP 동작원리>

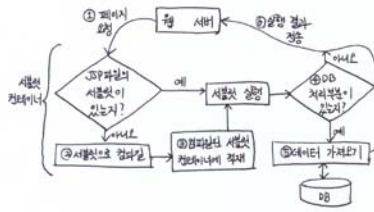


<내용 정리 3>

• (파워명-jsp.java) Java 코드의 구조

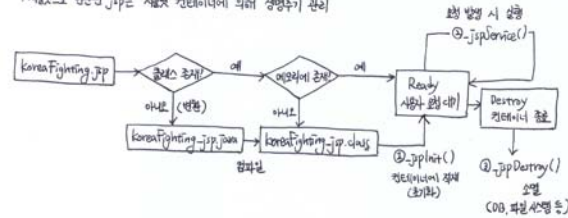
```
package 패키지;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
...
public void _jspInit(){
}
public void _jspDestroy(){
}
public void _jspService (...){
}
```

• JSP의 동작 구조



• 서블릿에서도 생명주기

: 서블릿으로 변환된 Jsp는 서블릿 컨테이너에 의해 생명주기 관리



2.

<내용 정리 4>

웹 프로그래밍 3주차

<스프릿 요소>

• 스크립트 요소: JSP 프로그래밍에서 사용되는 문법의 표현 형태 (선언문, 스크립트, 표현식, 주석)

• 선언문 (Declaration)

- JSP 페이지 내에서 문법으로 HTML 코드를 생성하는 데 사용될 변수나 메서드를 선언

- 문법 <%! 멤버 변수 및 메서드를 선언하는 영역 %>

- 선언문에 선언된 변수: JSP 코드가 실행될 때로 변수와 함께 서블릿 클래스의 멤버 방법으로 변환

- 선언문에 선언된 메서드: 일반적인 형태의 메서드로 변환

• 스크립트 (Scriptlet)

- JSP 코드가 서블릿으로 변환되고 요청될 때 ~~컴파일러~~ `_jspService()` 안에 선언되는 요소

- 문법 <% 지역 변수 선언, for, while, if 등... %>

• 표현식 (Expression)

- 동적으로 변수를 출력하거나 메서드의 결과값을 브라우저에 출력

- 스크립트 코드 내에서 out 이라는 내장객체를 통해 브라우저에 출력

- 세미콜론을 포함하지 않음.

- 문법 <% = 변수 혹은 메서드 %>

• 주석 (Comment)

- HTML 형태의 주석 <!-- 주석 -->

- JSP 형태의 주석 <% -- 주석 %> = 변수 혹은 메서드 %> - %> → 브라우저에서 확인 가능

- 스크립트 코드의 주석 <% /% 주석 (여러 줄) %> /% %> } 브라우저에서 보이지 않음
<% // 주석 (한 줄) %>
<% = 변수 혹은 메서드 /% 주석 %> /% %>

1

<내용 정리 5>

< 폼 태그 >

- * 폼 : 사용자와 웹 애플리케이션이 상호 작용하는 기술로, 사용자가 입력한 모든 데이터를 한 번에 웹 서버로 전송하고 처리 결과에 따라 다른 웹 페이지를 보여줌
- * 폼을 구성하는 태그의 종류 : form (화장위 태그), input (입력 받는 태그), select (선택 선택), textarea (여러 줄 입력)
- * <form> 태그
 - 선택으로 소시고 사용자가 다양한 장치를 입력하는 영역을 포함하는 최상위 태그
 - form 속성 : "값1" [속성2="값2" ...]
 - // 다양한 입력 태그 <input> <select> <textarea>
 - </form>
 - 속성 : action (데이터를 받아 처리하는 웹 페이지 URL 설정), method (전송되는 HTTP 방식 설정), name (폼 이름)
- * <input> 태그
 - input 속성 : "값1" [속성2="값2" ...]
 - 속성 : name (식별자), value (초값), type (text, radio, checkbox, password, hidden, file, button, reset, submit)
- * <textarea> 태그
 - 여러 줄의 텍스트를 <textarea> </textarea> 사이에 작성. 태그와 프라반기 그대로 출력
 - textarea 속성 : "내비 값" rows = "줄이 값"
 - 속성 : name (식별자), cols (열 크기), rows (행 크기)
- * <select> 태그
 - select 속성 : "값1" [속성2="값2" ...]
 - option 속성 : "값1" </option>
 - </select>
 - select 태그의 속성 : name (식별자), size (현재에 표시할 항목의 개수), multiple (다중 선택)
 - option 태그의 속성 : value (현재의 값), selected (초값으로 설정), disabled (강제 비활성화)

1

<내용 정리 6>

- * 폼 데이터 처리 과정



- * request 내장 객체
 - 웹 브라우저가 서버로 보낸 요청에 대한 다양한 정보를 담고 있음
 - getParameter(): 요청 파라미터의 값 얻기


```
String 변수 = request.getParameter("요청 파라미터 이름");
```
 - setCharacterEncoding(): 인코딩 설정


```
request.setCharacterEncoding("인코딩 방식");
```
 - getParameterValues(): checkbox 등의 입력 항목에서 동일한 이름의 파라미터 값을 String[] 형태로 받음
 - getParameterNames(): 입력 항목 전체의 파라미터 이름을 Enumeration 형태로 받음
 - hasNextElement(): Enumeration 요소가 있으면 true, 없으면 false
 - nextElement(): Enumeration 요소를 반환

2

<내용 정리 7>

<Page 자식 태그>

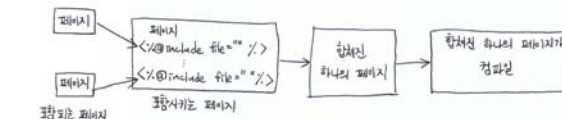
- 자식 태그: JSP 페이지가 서버로 클라이언트로 변환될 때 JSP 페이지를 어떻게 처리할 것인지 관련 정보 JSP 컨테이너에 지시하는 메타태그
 - page 태그: `<%@page ...%>`: JSP 페이지에 대한 정보 설정
 - include 태그: `<%@include ...%>`: JSP의 독립 영역에 다른 문서 포함
- page 자식 태그
 - 현재 JSP 페이지에 대한 정보 설정
 - 일반적으로 JSP 페이지의 최상단에 선언
 - `<%@page contentType="text/html" charset="UTF-8"%>`
- page 자식 속성
 - info: 현재 JSP 페이지 설명
 - language: 사용할 프로그래밍 언어 (ex. java)
 - contentType: 콘텐츠 타입 (MIME-type)
 - pageEncoding: 문자 인코딩
 - import: JSP 페이지에서 사용할 자바 클래스
 - session: 세션 사용 여부
 - buffer: 출력 버퍼 크기
 - autoFlush: 출력 버퍼의 플러시 제어
 - isThreadSafe: 멀티스레드 사용 여부
 - errorPage: 오류가 발생했을 때 보여줄 JSP 페이지 설정 (파일명)
 - isErrorPage: 현재 JSP 페이지가 오류 페이지인지
 - trimDirectiveWhitespaces: 페이지 소스 코딩 최면에서 JSP 문법으로 만들어지는 공백 제거

<Include 자식 태그>

- include 자식 태그
 - 현재 JSP 페이지의 특정 영역에 외부 파일(HTML, JSP, 텍스트 파일)의 내용을 포함하는 태그
 - JSP 페이지의 이식성을 선언 가능
 - `<%@include file="파일명"%>`
 - file 속성: 현재 JSP 페이지에 포함할 외부 파일의 상대 경로

<내용 정리 8>

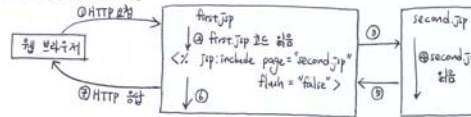
- include 자식 태그 처리 과정



<Include 액션 태그>

- 액션 태그
 - JSP 페이지에서 페이지와 페이지 사이를 제어
 - 다른 페이지의 실행 결과 내용을 현재 페이지에 포함
 - 자바 빈즈 등의 다양한 기능 제공
- include 액션 태그
 - include 자식 태그와 유사
 - 포함할 수 있는 외부 파일: HTML, JSP, 서버상 페이지 등
- include 액션 태그 종류
 - forward, include, webBean, setProperty, getProperty, param

- include 액션 태그 처리 과정



- include 액션 태그 문법

<규칙 1>
`<jsp:include page="호출 URL" flush="true"/>`
 사용자의 입력값을 request.getParameter()로 가져옴

<규칙 2>
`<jsp:include page="호출 URL">`
`<jsp:param name="" value="" />`
`</jsp:include>`
 param 액션 태그의 값을 request.getParameter()로 가져옴

- include 액션 태그 속성

- page: 현재 JSP 페이지에 포함할 외부 파일의 상대 경로
- flush: 설정한 외부 파일을 제어 시점 시 현재 JSP 페이지가 지금까지 출력된 뒤에 저장된 결과로 처리 (기본값: false)
 → true로 설정하면, 출력 버퍼에 저장된 내용을 웹 브라우저에 출력하고 출력 버퍼 비움.

<내용 정리 9>

• Include 액션 태그 vs Include 자식 태그

	Include 액션 태그	Include 자식 태그
문법	<code><jsp:include page="상대경로" /></code>	<code>< %@include file="상대경로" %></code>
처리 시간	요청 시	번역 시
가능	별도의 파일로 요청 처리 흐름을 이동	현재 페이지에 삽입
데이터 전달 방법	request와 내장 객체나 param 액션 태그 이용	페이지 내에 변수 선언
문법	화면 디자인상의 일관성 조율화	각각의 JSP 형 페이지에서 공통으로 사용되는 코드
기타	동적 페이지에 사용	정적 페이지에 사용

3

<내용 정리 10>

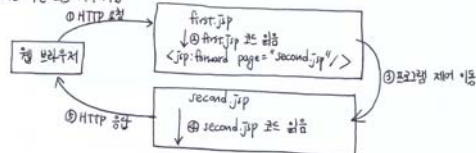
웹 프로그래밍 6주차

< forward 액션 태그 >

• forward 액션 태그

- 현재 JSP 페이지에서 다른 페이지로 이동하는 태그
- JSP 컨테이너는 현재 JSP 페이지에서 forward 태그를 만나면 그 전까지 출력 버퍼에 저장되어 있던 내용을 삭제하고 forward 액션 태그에 설정된 페이지로 프로그램의 제어 이동한다.

• forward 액션 태그 처리 과정



• forward 액션 태그 문법

<예제 1>
`<jsp:forward page="로컬 URL" />`
 <예제 2>
`<jsp:forward page="로컬 URL">`
`<jsp:param name=" " value=" " />`
`</jsp:forward>`

• forward 액션 태그 속성

- page: 현재 JSP 페이지의 이름 페이지에 해당하는 외부 파일의 상대 경로

< 자바 빈스 >

• JSP 페이지의 문제점

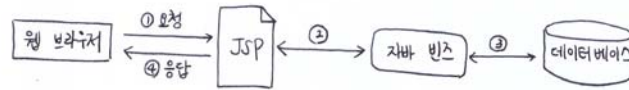
- JSP에 HTML을 제외한 다른 많은 JSP 코드가 있어 JSP 페이지를 이해하기 어려움
- 코드 재사용이 어려움

• 자바빈즈를 쓰는 이유

- JSP 안의 수많은 자바 코드를 담당했던 일들을 독립적으로 처리
- 계산이나 자료 처리를 담당하는 자바 코드 분리하여 작성
- JSP 코드가 간결해짐
- 동적 컨텐츠 개발을 위해 자바 클래스로 로직 작성

<내용 정리 11>

• 자바 빈 처리 과정



- 자바 빈 패키지 생성: 프로젝트/src/main/java 위치에 생성
 프로젝트/Java Resources / "src/main/java" / 패키지 } 2곳에서 생성된 패키지 확인 가능
 프로젝트/src/main/java / 패키지

• 자바 빈즈 코드 작성

- 정보를 저장하는 변수는 모두 private 으로 선언
- 변수의 값을 저장하는 setter() 메소드 public 으로 선언
- 변수의 값을 가져오는 getter() 메소드 public 으로 선언

• useBean 액션 태그

- 인스턴스 (객체) 생성

<jsp:useBean id="객체 이름" class="클래스 이름(패키지 경로 포함)" scope="빈즈 클래스 범위" />

- scope 유형: page (기본값, 해당 JSP 페이지 내부), request (요청 처리하는 동안), session (최초 접속 ~ 세션 종료), application (어플리케이션이 살아있는 동안)

• setProperty 액션 태그

- 빈의 변수에 값 저장 → setter() 역할

<jsp:setProperty name="객체명" property="변수명" value="값" />
 input 태그들의 name과 클래스의 변수(필드) 명이 일치하면
 <jsp:setProperty name="*" property="*" />
 로 값을 일괄 저장 가능

• getProperty 액션 태그

- 빈의 변수 값을 가져옴 → getter() 역할

<jsp:getProperty name="객체명" property="변수명" />

학습성찰

학습내용이해도

91%

<p>학습활동돌아보기 (좋았던점, 아쉬운점)</p>	<p>- 한아림</p> <p>오늘은 그간 웹 프로그래밍 스터디에서 공부한 내용을 복습하는 시간을 가졌다. 이전 스터디에서 공유하고 배웠던 내용들을 다시 복습하면서, 그동안 내가 이해하지 못했던 부분들이나, 미처 파악하지 못했던 내용들을 다시 한번 깊이 이해할 수 있었다. 또한, 기억하지 못했던 내용들도 다시 한번 상기시키면서 이전에는 이해하지 못했던 개념들이 이제는 더욱 명확해졌음을 느낄 수 있었다. 이러한 복습을 통해, 내가 스터디에서 배웠던 내용들을 더욱 확실히 이해하고, 습득할 수 있게 되었다. 이를 통해 웹 프로그래밍에 대한 이해도가 높아지며 앞으로 더 나은 학생이 되기 위해 노력하겠다는 다짐을 할 수 있었다. 마지막으로 스터디원들과 함께 공부하고, 서로 배움을 나누는 것이 이렇게 좋은 결과를 가져왔다는 것에 대해 감사한 마음을 느끼며, 앞으로도 더욱 열심히 공부하며 함께 성장해 나가고 싶다는 생각이 들었다.</p> <p>- 손효림</p> <p>오늘은 중간고사를 대비해 반 학기동안 배운 내용을 복습하였다. 특히 헛갈리던 자바 빈즈의 패키지와 클래스를 생성하고 코드를 작성하는 과정을 복습하고 useBean 액션 태그의 id, class, scope 속성의 설명과 객체를 언제까지 유지할 것인지에 따른 scope 유형 page, request, session, application에 대해 구체적으로 학습하였다. setter() 메소드와 getter() 메소드는 public으로 선언하고 변수는 private로 선언한다는 것을 복기해 절대 잊지 않을 것 같아서 유익하였다. 또한 getProperty가 빈의 값을 가져오면 setProperty에 빈의 값을 저장한다는 것이 헛갈렸는데 다시 짚어보며 확실히 체득할 수 있었다.</p> <p>- 이경화</p> <p>오늘의 스터디는 중간고사 시험범위에 대한 총복습을 다룬 시간이었다. 웹프로그래밍 개요부터 JSP 동작원리, 스크립트 요소, 폼태그, JSP 제어문, JSP 지시자, 액션태그, 자바 빈즈에 이르기까지, 중간고사에서 출제될 범위를 전반적으로 복습할 수 있었다. 이번 스터디에서 좋았던 점은, 스터디 시간이 충분히 주어진 점과 스터디원들과의 대화를 통해 이전에 공부했던 내용을 되새기며, 부족한 부분을 채울 수 있었던 점이다. 또한, 스터디 진행 중에는 참여자들끼리 서로 의견을 나누며, 생각을 공유하는 시간이 충분히 있었기 때문에, 개인적으로 이해하기 어려운 부분도 함께 이해할 수 있었다. 하지만, 스터디에서 아쉬웠던 점은 딱히 없었다. 전반적으로, 이번 스터디는 중간고사 시험범위를 복습하는 데에 매우 유익한 시간이었다. 스터디를 통해 부족한 부분을 채우고, 복습을 통해 시험에 대비할 수 있는 좋은 기회였다. 앞으로도 유용한 스터디가 계속해서 이루어질 것이 기대된다.</p>
	<p>다음학습계획</p> <p>일정 : 5월 8일 15:00~17:45 (50주년기념관 517호)</p>

2023년 1학기 스터디그룹 주간학습보고서