**Web개발\_Day46\_노트정리**

**설계 순서**

ㆍ바람직한 설계의 단계

1. 요구사항 확인
2. 기능/디자인
3. 기능추가, 완성도 높이기
4. 버그테스트
5. 배포, 서비스

* 요구사항에 맞는 기능들을 먼저 구현
* 단계별로 점진적으로 발전시켜 나감
* “버전관리” (=형상관리)
* V1.0.0 → v1.0.1 → v1.0.2 …등

↘→ v1.1.0 …등

**Form 태그의 action**

ㆍaction으로 목적지를 결정하지 않으면 post/get방식은 default로써 해당 페이지로 부른다.

ㆍGET/POST 에서의 500번대 에러발생의 경우!

* 근본 원인에 대한 문제 파악
* <% … %> 안에 코드를 먼저 실행하려는데 null인 경우 등
* “우선순위”를 확인하기

1. GET으로 현재 페이지를 요청
2. 현재 페이지가 브라우저에 출력
3. POST로 계산기능을 요청
4. 작업한 대로의 예상되는 결과가 브라우저에 출력

**개념정리, 제어문의 기능 분류**

ㆍ제어문의 기능에 따른 분류

* if문 : ‘분기점’을 기준으로 코드가 나뉠 경우
* switch문 : 버튼식 수행일 경우
* while문 : 반복횟수가 불분명할 때
* for문 : 반복횟수가 명확할 때 ( 컬랙션의 크기에 따른 반복, 일정횟수 반복)

ㆍ위와 같이 기능의 목적을 알고 사용해야 한다

* GET/POST 의 구분도 동일하게, 확실하게 분류하고 사용하여야 한다.

**JSP 기초 시작하기**

ㆍ <!-- --> : 설명 등을 보여주기 위한 주석

ㆍ<%-- --%> : 사용자에게 보이지 않는 주석

ㆍ지시어 : <%@로 시작한다.

* Page : language, contentType, import, errorPage + isErrorPage 설정 등이 가능
* errorPage, isErrorPage는 아래 참고
* include : 다른 페이지의 내용을 포함시킬 수 있음
* .jsp 파일들을 포함시킨 후 컴파일 하기 때문에 정적인 페이지들에 유리
* Taglib : “커스텀태그”

ㆍ에러잡기

* <%@page 페이지 지시어에서... errorPage=”에러발생시 이동할 페이지”
* 에러발생시 표시할 페이지
* isErrorPage=”true” 로 해당 페이지가 에러를 다루는 페이지라는 것을 표기한다
* default값은 false로 자동 지정되어 있다.

**액션 <jsp:OOO />**

ㆍ <jsp: ~ > </jsp:~> 와 같은 형태를 이룬다.

* <jsp:~> 의 태그바디 안에 <jsp:param value=”” name=””/> 을 작성할 수 있는데, 값을 전달 가능하다.

ㆍ액션태그 <jsp:~>의 종류

1. Include : .jsp파일들을 실행시점에서 호출

* Include 지시어에 반해, include액션태그는 동적인 페이지들에 유리하다

1. useBean(+setProperty+getrProperty) : import + new 생성자() 의 2가지 기능을 한다

* Bean은 객체를 의미하는데, 객체는 ‘기능’을 의미한다
* New 생성자() 를 의미하되, 객체가 존재한다면 다시 생성하진 않는 기능을 제공한다.

1. Forward : [제어권]

* Forward할 페이지로 제어권을 완전히 넘겨준다
* 제어권을 갖는 페이지만 출력이 되기 때문에, forward를 사용하는 페이지는 출력이 되지 않는다.

**표현식**

ㆍ <%! %> : 선언식.

* 변수 메소드 등을 선언할 때 사용

ㆍ<%= %> : 표현식

* 안에 적은 문자를 출력해주는 기능
* Out.println(); 안에 들어가는 문장을 out.println() 대신해서 출력해주는 것과 동일하므로 표현식 내부에는 ; 를 사용하지 않는다.

ㆍ<% %> : 스크립트릿

* 자바를 온전히 사용할 수 있게 하는 공간을 제공한다.