**JSTL(JSP Standard Tag Library )과 일반 스크립트릿(자바구문)의 비교  
스크립트릿(자바구문),**

**익스프레션(표현식) ,**

**선언문 같은 JSP 스크립팅 엘리먼트를 사용하는 태그를 구현할 수 있다.**

**그러나 페이지 내에 프로그램 소스 코드를 직접 작성하게 되면 소프트웨어 관리 태그를 매우 복잡하게 하는 경향이 있다.**

< % if ( a == 0 ){ % >  
< %= a % >  
< %} %>

**위와 같이 복잡해 보이는 코드를 아래와 같이 줄일 수 있다.**

**< c:if test="${a==0}">  
   < c:out value="${a}">  
< /c:if>**

**JSP 표준 태그 라이브러리[JSP Standard Tag Library]**

**JSP 태그를 라이브러리 형태로 모든 프로그래머가 공통으로 사용하도록 모아 놓은 것이다.**

**이와 대조적으로 프로그래머가 재사용하기 위해 미리 정해 놓고 사용하는 JSP 커스텀 태그도 있다.**

**JSTL (JSP Standard Tag Library=JSP표준 태그 라이브러리) 이 제공하는 태그의 종류 5 가지**

**jstl 1.1 은 다음과 같이 5가지 종류의 태그를 제공한다. ( jstl.jar )**

|  |  |  |  |
| --- | --- | --- | --- |
| **라이브러리** | **하위기능** | **접두어(prefix)** | **관련URl** |
| **코어(core)** | **변수지원 흐름제어 URL처리** | **c** | **http://java.sun.com/jsp/jstl/core** |
| **XML** | **XML코어 흐름제어 XML변환** | **x** | **http://java.sun.com/jsp/jstl/xml** |
| **국제화** | **지역 메시지형식 숫자,날짜 형식** | **fmt** | **http://java.sun.com/jsp/jstl/fmt** |
| **데이터베이스** | **SQL** | **sql** | **http://java.sun.com/jsp/jstl/sql** |
| **함수** | **콜렉션 처리 String 처리** | **fn** | **http://java.sun.com/jsp/jstl/functions** |

**이중에서 코어라이브러리와 함수 라이브러리를 많이 사용한다**

**그리고 출력결과 화면에 보여줄 메시지를 체계적으로 관리하기 위해서**

**국제화 라이브러리를 종종 사용하기도 한다**

**접두어는 JSP페이지에서 커스텀태그를 호출할 때 사용한다**

**다른 접두어를 사용하더라도 상관은 없다**

**관련 URI는 JSTL 이 제공하는 커스텀 태그를 구분해 주는 식별자를 통해서**

**JSP페이지에서 사용할 커스텀 태그 라이브러리를 선택할 수 있다**

**JSTL를 사용하기 위한 환경설정**

**탐캣5.0은 기본적으로 JSTL을 지원하지 않으며**

**JSTL API 를 제공하는 jar 파일을 별도로 구해야 한다**

**jstl.jar**

**standard.jar 를**

**/WEB-INF \ lib 에 복사**

**코어태그(core)**

**코어 태그 라이브러리는 if-else나 변수의 지정 등과 같이 논리적인 처리를 위해 사용되는 스크립트 코드를 대체하기 위한 태그를 제공하는데, 제공하는 태그는 아래와 같다.**

**Core 태그**

|  |  |  |
| --- | --- | --- |
| **기능 분류** | **태 그** | **설 명** |
| **변수 지원** | **set** | **JSP에서 사용될 변수를 설정한다.** |
| **remove** | **설정한 변수를 제거한다.** |
| **흐름 제어** | **If** | **조건에 따라 내부 코드를 수행한다.** |
| **choose, when** | **다중 조건을 처리할 때 사용한다.** |
| **forEach** | **콜렉션이나 Map의 각 항목을 처리할 때 사용한다.** |
| **forTokens** | **구분자로 분리된 각각의 토큰을 처리할 때 사용한다.** |
| **URL 처리** | **import** | **URL을 사용하여 다른 지원의 결과를 삽입한다.** |
| **redirect** | **지정한 경로로 리다이렉트한다.** |
| **url** | **URL을 재작성한다.** |
| **기타 태그** | **catch** | **예외 처리에 사용한다.** |
| **out** | **JspWriter에 내용을 알맞게 처리한 후 출력한다.** |

**코어 태그 라이브러리를 사용하려면 JSP 페이지에 다음과 같이 taglib 디렉티브를 추가해 주어야 한다.**

|  |
| --- |
| **<%@ taglib prefix=”c” uri=”**[**http://java.sun.com/jsp/jstl/core**](http://java.sun.com/jsp/jstl/core)**” %>** |

**여기서 prefix 속성에 명시된 값은 코어 태그 라이브러리를 호출할 때 사용할 접두어를 의미한다.**

**변수는 지원해주는 tag 는 set , remove**

**\* set 태그는 EL변수나 EL변수의 프로퍼티의 값을 지정할 때 사용한다**

**page, request, session , application 의 영역에 저장되어 있는 값을 처리할 때 사용**

**\* 객체는 <jsp:useBean> 태그와 <jsp:setProperty> 태그를 사용하는 것과 마찬가지로**

**예) 변수지정 <c:set>**

|  |
| --- |
| **<c:set var="name" value="expression" scope="page"/>** |

**scope=”page” 일 경우 는 디폴트이기 때문에 생략가능,**

**remove 할때는 scope=”page”를 반드시 지정한다**

**설명)**

|  |
| --- |
| **<c:set var="varName" scope=”session” value="someValue"/>**  **변수이름 변수를 저장할 영역지정 변수의값**  **session**  **request**  **page ( 디폴트 )**  **application** |

**value 속성을 지정하지 않고 태그 몸체에 값을 입력할 수도 있다**

|  |
| --- |
| **<c:set var="varName" scope=”request” >**  **someValue**  **</c:set>** |

**remove 태그는 set 으로 지정한 변수를 삭제할 때 사용**

**<c:remove> 태그를 사용할때는 아래 예처럼 스코프를 명확히 명시해야 한다:**

|  |
| --- |
| **<c:remove var="varName" scope="application”/>**  **<c:remove var=" varName" scope="request/>** |

**remove 태그 사용시 주의 할점은**

**삭제할 변수의 scope를 올바르게 지정하지 않으면 변수가 제거되지 않는다**

**출력하고,지정하고,삭제하고 에러처리 하기 위해 다음과 같은 태그를 사용한다.**

**<c:out>, <c:set>, <c:remove>, and <c:catch>**