



JSTL

JSTL 의 4가지 종류

- JSTL (JSP Standard Tag Library)
 - 개발자들이 개별적으로 만들어 쓰던 태그 중 자주 사용되는 태그를 표준으로 만들어서 제공
- core 태그 - 프로그램에서 많이 사용되는 기본 태그 제공
- fmt 태그 - 다국어 지원, 로케일 제공, 메시지 처리, 숫자, 날짜 포매팅
- 함수
- sql 태그
- xml 태그

core - 모든 사이트 주요 사용 태그

fmt , 함수 - 빈도는 낮지만 가끔 사용

sql, xml - 거의 사용하지 않음

JSTL 사용

1. 다운로드

- <http://tomcat.apache.org/taglibs/standard/>
- <http://mvnrepository.com>

2. JSTL 관련 라이브러리 파일을 프로젝트에 사용할 수 있도록 임포트한다.

- WebApplication 폴더 밑 WEB-INF / lib 폴더에 저장

3. JSP 페이지 작성 시 JSTL 관련 정보를 taglib 지시자를 이용해서 알린다.

- 사용 예

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<c:out value="test" />
```

JSTL 사용

1. 다운로드

-

| 라이브러리 | 접두어 | 관련URI |
|--------|-----|----------------------------------------|
| 코어 | c | http://java.sun.com/jsp/jstl/core |
| XML | x | http://java.sun.com/jsp/jstl/xml |
| 국제화 | fmt | http://java.sun.com/jsp/jstl/fmt |
| 데이터베이스 | sql | http://java.sun.com/jsp/jstl/sql |
| 함수 | fn | http://java.sun.com/jsp/jstl/functions |

2. JSTL

-

3. JSP

-

록 임포트한다.

용해서 알린다.

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<c:out value="test" />
```

core - set 태그

1. 형식

- `<c:set var="" value="" [scope=""] />`

2. 공유 영역에 변수 설정 시 사용

3. 속성 설명

- var : 설정할 변수명
- value : 설정할 값
- scope : 변수의 사용 영역(page, request, session, application)

기본설정 영역 : page

4. 사용 예

```
<c:set var = "cnt" value = "1" scope = "page" />
```

core - out 태그

1. 형식

- `<c:out value="" [escapeXml="true/false"] [default=""] />`

2. 출력 전용 태그

3. 속성 설명

- value : 출력할 내용을 지정
- escapeXml :
 - true 일 경우 : 출력내용에 태그가 있을 경우 문자값으로 변경처리
 < : < > : > & : & ' : ' " : "
- default
 - value 속성의 값이 없을 경우 디폴트로 출력할 내용 지정

core - out 태그

4. 사용 예

<c:out value =“JSTL” /> ➔ JSTL

<c:out value = "\${ msg }" default="값 없음" /> ➔ 값 없음

<c:out value =“<hr />” default=“값 없음” /> ➔ <hr />

```
<c:out value = “내용에 hr 태그 적용 <hr />” default = “값 없음”  
escapeXml = “false” />
```

➔내용에 hr 태그 적용

core - if 태그

1. 형식

- `<c:if test="조건">`
조건이 참일 경우 실행할 내용
`</c:if>`

2. 조건에 따라 내용을 실행

3. 속성 설명

- test : 참과 거짓을 판단하는 조건을 명시

4. 사용 예

```
<c:if test="${5 == 5}">  
    5 == 5 와 같습니다.  
</c:if>
```


core - choose 태그

1. 형식

- `<c:choose>`
 - `<c:when test="">`
조건이 참일 경우 실행할 내용
 - `</c:when>`
 -
 - `<c:otherwise>`
when 절이 모두 거짓일 경우 실행할 내용
 - `</c:otherwise>`
 - `</c:choose>`

2. 다중 조건 처리 태그

3. 속성 설명

- test : 참과 거짓을 판단하는 조건을 명시

core - choose 태그

4. 사용 예

```
<c:choose>
  <c:when test="${param.type == 'S'}">
    당신은 슈퍼관리자
  </c:when>
  <c:when test="${param.type == 'U'}">
    당신은 일반 사용자
  </c:when>
  <c:otherwise>
    올바르지 않은 경로로 접근했습니다.
  </ c:otherwise >
</c:choose>
```

core - forEach 태그

1. 형식

- `<c:forEach var="" items="" [varStatus=""]>`

반복 시 처리 할 문장

`</c:forEach>`

배열 또는 컬렉션 객체를 반복할 경우 사용

- `<c:forEach var="" begin="" end="">`

반복 시 처리 할 문장

`</c:forEach>`

지정된 회수 만큼 반복할 경우 사용

2. 반복문 처리 태그



core - forEach 태그

3. 속성 설명

- var : 반복문에서 사용 할 변수명 지정
- items : 반복시킬 대상 객체 지정 (배열 또는 컬렉션)
- varStatus : 변수명, 반복과 관련된 기본 정보를 담고 있는 객체
 - 1) first : 반복문의 처음인지 판단, 참과 거짓을 반환
 - 2) last : 반복문의 마지막인지 판단, 참과 거짓을 반환
 - 3) index : 반복의 처음에 0을 반환, 반복이 진행될때마다 1씩 증가
 - 4) count : 반복의 처음에 1을 반환, 반복이 진행될때마다 1씩 증가
- begin : 반복문 범위의 초기 값 설정
- end : 반복문 범위의 마지막 값 설정

core - forEach 태그

4. 사용 예

```
for(int i=1; i<=10; i++) {  
    System.out.println(i);  
}  
<c:forEach var="i" begin="1" end="10">  
    ${i}  
</c:forEach>  
  
String [] strArr = {"1", "2", "3"};  
for(String s : strArr) {  
    System.out.println(s);  
}  
request.setAttribute("strArr", strArr);  
<c:forEach var="s" items="${strArr}">  
    ${s}  
</c:forEach>
```

자바코드
비교

core - redirect 태그

1. 형식

- `<c:redirect url="페이지 주소" />`

2. `response.sendRedirect("페이지 주소");` 을 대체

3. 속성 설명

- `url` : 넘어갈 페이지 주소

4. 사용 예

```
<c:redirect url="http://www.naver.com" />
```

core – import 태그

1. 형식

- `<c:import url="페이지 주소" />`
- `<c:import url="페이지 주소">`
`<c:param name="id" value="test" />`
`</c:import>`

2. 다른 페이지의 실행 결과를 내 페이지로 포함시킴

3. 속성 설명

- url : 결과 내용을 포함할 페이지 주소

4. 사용 예

```
<c:import url="http://www.naver.com" />
```

자주 사용되는 함수

| 함수 | 설명 |
|-------------------------------|-----------------------------------------------------------|
| fn:length(obj) | List와 같은 Collection인 경우 저장된 개수를 반환 문자열일 경우 문자열의 길이를 리턴 |
| fn:substring(str, idx1, idx2) | str의 내용에서 idx1 위치에서 idx2 - 1 사이의 문자열 반환 |
| fn:trim(str) | str 좌우의 공백문자를 제거 |
| fn:replace(str, src, dest) | str에 있는 src를 dest로 변환 |
| fn:indexOf(str1, str2) | str1에서 str2가 위치한 인덱스를 구한다. |
| fn:startsWith(str1, str2) | str1이 str2로 시작할 경우 true를, 그렇지 않을 경우 false를 리턴한다. |
| fn:endsWith(str1, str2) | str1이 str2로 끝나는 경우 true를, 그렇지 않을 경우 false를 리턴한다. |
| fn:contains(str1, str2) | str1이 str2를 포함하고 있을 경우 true를 리턴한다. |

- 사용 예

```
${fn:length("fn test")}
```

```
${fn:substring("fn test", 5, 6)}
```

```
${fn:substring("fn test", 5, -1)}
```

```
${fn:replace("fn test", "fn", "test")}
```

```
${fn:indexOf("fn test", "es")}
```

```
${fn:startsWith("fn test", "fn")}
```

```
${fn:endsWith("fn test", "est")}
```

```
${fn:contains("fn test", "est")}
```