

el과 jstl



- 요청값을 처리할 때, 효과적인 el을 활용하여 프로그래밍 할 수 있다.
- el을 통한 산술 및 논리 연산 처리를 할 수 있다.
- el이 제공하는 기본 객체의 기능별 구별을 할 수 있다.
- jstl의 개념과 기본적인 활용 방법을 안다.

- jstl의 개념과 기본적인 활용 방법을 안다.
- jstl에서 page scope[page, request, session, application]별로 데이터를 jsp단에서 저장할 수 있고, 활용할 수 있다.
- jstl을 통해서 반복문, 조건문을 처리할 수 있다.
- jstl에 사용되는 날짜, 숫자에 대한 format형식을 효과적으로 처리할 수 있다.

### 생각해 봅시다 🖁

- jsp에서 사용한 스클립트릿<%%>, 익스프레션<%=%> 가독성과 사용에 어려움이 있지 않는지?
- el나 jstl을 활용하면서 변수를 지정하는데 어떤 범위에 대한 처리를 할 수 있을까?
- session값이나 요청값에 대한 type casting,
   null 은 el에서 해결해 줄 수 있는가?



- EL(Expression Language) 표현언어는
  JSTL(Jsp Standard Tag Library)1.0 규약에
  소개된 내용으로 jsp 2.0 버젼부터 표현 언어로
  포함되어 있다.
- 새로운 스크립트 언어로 jsp 기본문법 보완
- 기능
  - jsp 의 네 가지 기본 객체가 제공하는 영역 속성 사용
  - \_ 집합 객체에 대한 접근 방법 제공
  - 수치연산, 관계 연산, 논리 연산자 제공
  - 자바클래스 메서드 호출 기능 제공
  - 표현언어 기본 객체 제공





- 기본 문법
  - \${표현식}
  - ex)
  - Person p = new Person(); p.setName("홍길동")
  - session.setAttribute("person", p );
    - <% Person</p> person=(Person)session.getAttribute("perso n");
    - if( person!=null){
    - <%=person.getName()%>
    - ==> \${person.name}





- 요청값 처리
  - String id=request.getParameter("id");
  - if(id!=null)
  - 필요한 type(숫자) 변환
  - ==> \${param.id}



■ 요청값 처리 기본 @@ 커피 프린스에서 @@ 구매할 커피 가격 갯수 [주문] 주문하신 커피는 @@@ 가격 @@@에 @@잔 으로 계산서 @@@원 발급합니다.

## 비교연산자 처리...

- \${ 값1 == 값2} => \${값1 eq 값2}
- \${ 값1!= 값2} => \${값1 ne 값2}(not equal)
- \${ 값1 < 값2} => \${값1 lt 값2}(less than)
- \${ 값1 > 값2} => \${값1 gt 값2}(greater than)
- \${ 값1 >= 값2 } ==> \${ 값1 le 값2}
- \${ 값1 <= 값2 } ==> \${ 값1 ge 값2}
- 요청값 ""(공백) null에 대한 처리..
- \${ empty 값} : 공백이거나 null일 때 true
- \${ not empty 값} : 공백이거나 null 아닐 때 true
- \${ 조건1 && 조건2} ==> \${조건1 and 조건2}
- \${!조건1} ==> \${not 조건1}

# request/session 객체 el로 처리:

- 요청값 저장이나 세션값 저장은 해당 4가지 scope범위에 대한 데이터를 el로 효과적으로 가지고 올 수 있다.
- request.setAttribute("name", "홍길동");
- ==> \${name}
- Product p = new Product();
- p.setPrice(5000); p.setName("바나나")
- session.setAttribute("prod", p);
- = ==>\${prod.price}, \${prod.name}

### el과 session 처리

- login
- ID : [ ]
- PASS: [ ] ==> id 님 환영합니다.
- [세션처리] 설정된 비번은 @@@
- ==> session Member에 id pass 할당하고.
- 다음 page에서 el를 이용하여 위와 같이 출력하게 처리.



- jsp 표준 태그 라이브러리
  - html로 지원하는 tag를 확장해서 jsp에서 개발자가 활용하는 태그를 만든 것을 커스텀 태그라고 한다.
     커스텀 태그 중에서 많이 사용되는 것을 모아서
     JSTL 이라는 규약을 만들고 이를 활용하고 있다.
  - 스크립트 코드를 사용할 때, 보단 간결하고 이해하기 쉬운 jsp 코드를 작성하게 도와 준다.
- 태그 종류
  - 코어(<c:XXX>): 변수, 흐름 제어, url 처리
  - 국제화(<fmt:XXX>): 숫자 및 날짜 형식 처리
  - XML(<x:XXX>)

## jstl 라이브러리 활용:

- jstl 라이브러리 다운
  - http://tomcat.apache.org/download-taglibs.cgi
- WEB-INF/lib 위치
  - jstl.jar, standard.jar
- jsp 상단
  - <%@ taglib prefix="c"
    uri="http://java.sun.com/jsp/jstl/core" %>
- 화면 확인
  - <body>
  - <c:set var="callMe" value="jstl호출되었습니다!!"/>
  - < h1> {callMe}</h1>

# 코어 태그:

- if~else나 변수의 지정 등과 같이 논리적 처리
- 태그 라이버러리
  - 변수 지원
    - <c:set> : jsp에 사용될 변수를 지정
      - var : 변수명, value : 변수에 매핑되는 값
      - ex) <c:set var="num01" value="25" />
    - <c:remove> : 설정한 변수 제거
  - 흐름 제어
    - <c:if> : 조석을 제어할 때, 활용
      - test: boolean 조건식(el)
      - ex) <c:if test="\${param.num01>=60}">합격</

# jsp에서 scope 범위:

- <c:set scope="속성값"/>
  - page : 현재 page까지
  - request : 요청 page까지
  - session: 브라우저 단위까지.
  - application : 서버 단위까지

### <c:set 객체할당

- 처리 프로세서
  - 객체 생성 :
    - request/session.attribute("key",객체);
    - Person p = new Person();
  - 객체 변수할당
    - <c:set var="obj01" scope="session/application"/>
      - value : \${key}, <%=p%>
      - ex) <c:set value="\${key}"/>
  - 객체의 메서드 활용
    - \${obj01.property} getName==> name
  - 메서드(property)를 통해서 필드 값의 변경.
    - <c:set target="\${obj01}" property="name" value="신길동"/> ==> obj01.setName("신길동")

# 확인예제

- Emp 객체 생성
  - ename:@@@
  - sal를 할당
- request에 attribute로 저장
- <c:set 이름 em01로 할당..
- 출력:
- 사원이름:@@@, 사원급여:@@@,
- 연봉A등급 여부(7000이상): true/false

### 숙제 및 확인예제 🖁

■ 거래 내역(2단계): a13\_jstlTarget.jsp 거래 내역(횟수:7)

물건명	가격	주문수량	계
딸기	5000	2	10000
오렌지	2000	5	10000
	저장		

- 초기에 없음, 입력하고 저장하는 다음 page부터 나타남.
- Product객체 <c:set 저장</li>
- 두번째 이후는 property 호출하여 저장처리.



## 숙제 및 확인예제 :

거래 내역(1단계)

물건명	가격	주문수량
[ ]	[ ]	[ ]

저장

- \${param.name}
- request.setAttribute 활용
- <c:set target 활용</p>



# 기본 예제:



# 이 장이 끝나면:



# 정리:



# 참고자료:

