

-el - <% %> <jsp:xxx />==> 자바문장 최소화. 태그 형태. 내부동작 자바코드자동변환방식

xxx.setxxx()

request.

session.

application

MemberDTO객체.setxxxx

<jsp:setProperty

<jsp:getProperty

<% out.println("jsp"); %>

<%= "jsp"%>

\${ "jsp" }

\${ 'jsp' }

\${ 문자열 논리값 정수 실수 null }

\${ 데이터 연산자 데이터 } ==> empty 연산자(null판단) , +=(문자열결합)

== | eq != | ne > | gt >= | ge

\${내장객체}

\${param.요청파라미터명}

\${paramValues.요청파라미터명[0]}

\${ i }

\${ pageScope.i }

\${ requestScope.i }

\${ sessionScope.i }

\${ applicationScope.i }

-jstl

jsp standard tag library

<% %> + <jsp:xxxxx /> +

<a href=

<c:set var="i" value="{j + 100}" />

<c:set var="i" value="<%=k + 100%>" />

<c:set var="i" value="100" />

\${i}

<c:remove var="i" />

<c:if test="t/f" >

here

</c:if>

<c:choose>

<c:when test="t/f" > here </c:when>

<c:when test="t/f" > here </c:when>

<c:when test="t/f" > here </c:when>

<c:otherwise> .. </c:otherwise>

</c:choose>

<c:forEach items="맵" var="i" >

\${i.key} - \${i.value}

</c:forEach>

<c:forEach items="배열이나 리스트" var="i" >

\${i.id} -> i.getId() 호출

</c:forEach>

<c:forEach items="맵" var="i" varStatus="변수명" >

?번째 반복 - \${i.key} - \${i.value}

</c:forEach>

varStatus-반복상태정보

index	반복 번호. 0부터 \${ 변수명.index}
count	반복 번호. 1부터 \${ 변수명.count}
first	반복 첫번째요소 \${ 변수명.first}
last	반복 마지막요소 \${ 변수명.last}
current	반복중인 요소 \${변수명.current.key} \${변수명.current.value.name}

<c:url var="mypage" value="http://localhost:9090/a/b3.jsp" />--> 이동/동작 없다

<a href= \${mypage}>

<form action

<img src

<jsp:include

<jsp:forward

<a href= \${mypage}>>

<a href= \${mypage}>>

<c:url var="mypage" value="http://localhost:9090/a/b3.jsp" />--> 이동/동작 없다

<c:redirect url="\${mypage}" />-->이동

<c:import url="\${mypage}" /> --> 포함

<c:param

<c:set var="i" value="100" /> 변수 정의

\${i} 변수 조회 출력

<c:out var="i" defaultValue="0"/> 변수 조회 출력



## 16장 jquery ajax – 638p

ajax 개념 – asynchronous javascript and xml 약자- xml 이용한 비동기적 자바 스크립트(JQUERY 간편) 개발 통신 방식

### <HTTP> 요청 응답 특징

1. 클라이언트가 요청
2. 요청한 파일(html, jsp, servlet등)을 클라이언트에게 응답
3. 요청했던 클라이언트의 브라우저 화면은 응답받은 파일로 변경
4. 요청 이전의 화면은 없어짐
5. 잦은 화면 변경의 경우 비효율적

http : 요청1- -응답1종료/-요청2-응답2(synchronous 통신)

jstl2.html	jstl2.jsp
<html>	<html>
<form action="jstl2.jsp"	<table>
<table>서버응답결과포함(텍스트)</table>	....

### SPA – SINGLE PAGE APPLICATION

- 1> 화면 전체 변경되기보다는 화면 일부만 변경 효율적
- 2> 로그인 처리 / 쇼핑목록 조회 처리 동시( 비동기적 )
- 3> 서버와 클라이언트 데이터 주고 받는다(현재 HTML 포함 일부데이터)
- 4> 일부데이터 이전 XML 형태 사용--> JSON 형태 사용
- 5> JAVA SCRIPT OBJECT NOTATION = 자바스크립트객체표현방식

### <SCRIPT>

```
var one = 100;

var ar = [100, 200, 300];

var obj = { id:'java' , name:"이자바", .... };

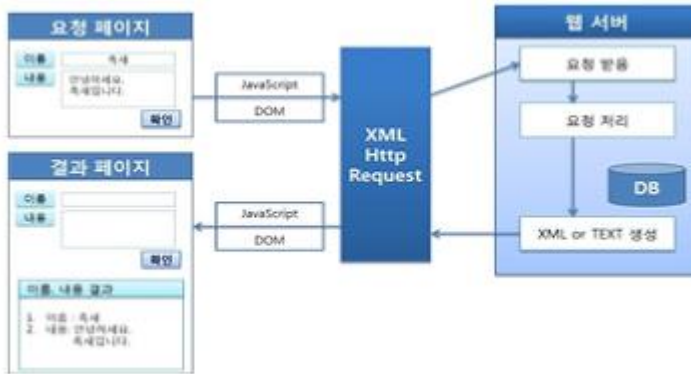
var ar = [100, { id:'java' , name:"이자바", .... }, 300];

var obj = { id:'java' , name:"이자바", hobby:['음악','영화'] };
```

### JSON 형태 규칙 { "변수명": "변수값", [], "NUM":"100", , , }

6> AJAX 서버와 클라이언트 데이터형태 약속-JSON – 모든 언어 주고 받는 데이터 타입

7> 비동기식이란 여러가지 일이 동시에적으로 발생한다는 뜻으로, 서버와 통신하는 동안 다른 작업을 할 수 있다는 의미



AJAX 클라이언트	AJAX 서버
<p><b>\$.ajax({설정사항})</b></p> <ul style="list-style-type: none"> <li>jQuery를 이용한 ajax 통신의 가장 기본적인 API</li> <li>주요속성 <ul style="list-style-type: none"> <li><b>data</b> : 서버에 전송할 데이터, key/value 형식의 객체</li> <li><b>dataType</b> : 서버가 리턴하는 데이터 타입 (xml, json, script, html)</li> <li><b>type</b> : 서버로 전송하는 데이터의 타입 (POST, GET)</li> <li><b>url</b> : 데이터를 전송할 URL</li> <li><b>success</b> : ajax 통신에 성공했을 때 호출될 이벤트 핸들러</li> </ul> </li> </ul> <p>- JQuery의 ajax 구현예</p> <pre>\$.ajax({   type: "post",           //http 요청방식   url: "serverPage.jsp", //서버측 페이지   dataType: "JSON",       //응답데이터 타입설정   success : function(data){ /*콜백함수*/     //응답 결과가 성공상태 코드로 반환되면 호출되는     콜백함수     error : function(data){ /*콜백함수*/     //응답 결과가 에러상태 코드로 반환되면 호출되는     콜백함수   }})</pre>	<p>SERVEPAGE.JSP</p> <p>요청 처리 응답 &lt;HTML&gt;태그 출력하지 않는다</p> <pre>String s = "{W"resultW" : W"로그인처리결과W", W"이름W" : W"값W" , , }"; &lt;c:set var="s1" value="&lt;%=s%&gt; /&gt; \${s1} or &lt;%=s %&gt; or &lt;% OUT.PRINTLN(s); %&gt;</pre>

서버 = 자바 "{W"...}"-->자동 json 변경 api

스프링 ajax

필요라이브러리세팅

json-simple.jar ==> webcontent\WEB-INF\lib	jquery-3.6.0.min.js ==> webcontent\폴더\*.js
---	---

\*.js--> java script 라이브러리

\*.jar -> 서블릿이나 jsp 라이브러리

```
{
  "address": "제주",
  "phone": "010-222-2222",
  "name": "회원1",
  "time": "2022년 03월 08일 14시 15분",
  "id": "ajax",
  "login": true,
  "email": "A@email"
}
```

사용자 name 여러명

```
JSONObject o1 = new JSONObject();
```

```
o1.put("one", "이자바");
```

```
JSONObject o2 = new JSONObject();
```

```
o2.put("one", "박자바");
```

```
JSONObject o3 = new JSONObject();
```

```
o3.put("one", "김자바");
```

```
JSONArray ar = new JSONArray();
```

```
ar.add(o1); ar.add(o2); ar.add(o3);
```

==>

```
[{"one", "이자바"}, {"one", "박자바"}, {"one", "김자바"}]
```

loginform3.jsp	loginresult3.jsp
alluserbtn 버튼 클릭하면 loginresult3.jsp ajax 호출한다 응답 결과를 alluserdiv 태그 테이블 형태로 출력한다	member_table 모든 회원정보 가져와서 MemberDAO selectMember() ArrayList --> MemberDTO-->JSONObject-->JSONArray --> 응답한다

json

JSON.parse()

JSON.stringify()

클라이언트코드/서버코드

17장

mvc

model1

model2

mvc-> front controller + mvc

18장 스프링 환경설정

1> 이클립스 툴 + spring 플러그인 기능 추가

2> sts 툴 설치=> 스프링 기능 이미 추가 이클립스

4>jdk 11 필요 --> jdk 1.8 변경

jdk 1.8 ----- 네이버 서버 (jdk 1.8)

-기본

eclipse 2000 6 + jdk 8

sts 4 + jdk 11



