

표현 언어와 JSTL

- 표현언어

표현언어로 표현 단순화하기

❖ 표현언어(EL: Expression Language)

- JSP 출력에 대한 부분을 쉽게 하기 위해 개발한 태그
- 형식
 - `${표현식}`

■ 예 : 표현식

`<%=expr%>`

`<%= "Hello"%>`

표현식(expression)

■ 예 : 표현 언어

`${expr}`

`${ "Hello" }`

표현 언어(Expression Language)

표현언어로 표현 단순화하기

❖ 01_el.jsp

```
<body>
  <!--표현 언어(EL, Expression Language : 이렇게 값을 가져오면 편하다
-->
  ${"Hello"} <br>
  <%= "Hello"%> <br> <!--스크립트릿-->
  <% out.println("Hello"); %> <br> <!--표현식( Expression)-->
</body>
</html>
```

표현언어로 표현 단순화하기

❖ 02_el.jsp

```
<body>
    정수형 : ${10} <br>
    실수형 : ${5.6} <br>
    문자열형: ${"성윤정"} <br>
    논리형: ${true} <br>
    null : ${null} <br>
</body>
</html>
```

표현언어로 표현 단순화하기

❖ EL

○ 연산자

종류	연산자
산술	+, -, *, / (or div), % (or mod)
관계형	== (or eq), != (or ne), < (or lt), > (or gt), <= (or le), >= (or ge)
조건	a ? b : c
논리	&& (or and), (or or), ! (or not)
null 검사	empty

`${5+2}`

`${3==3}`

`${empty input}`

연산자를 포함하는 EL 식

표현언어로 표현 단순화하기

❖ 03_el.jsp

```
<body>
  \${5+2} : ${5+2} <br>
  \${5/2} : ${5/2} <br>
  \${5 div 2} : ${5 div 2} <br>
  \${5 mod 2} : ${5 mod 2}<br>
  \${5 > 2} : ${5 > 2}<br>
  \${2 gt 10} : ${2 gt 10}<br>
  \${(5 > 2) ? 5 : 2} : ${ (5 > 2) ? 5 : 2}<br>
  \${(5 > 2) || (2 < 10)} : ${ (5 > 2) || (2 < 10)}<br>
<%
  String input=null;
%>
  \${empty input} : ${empty input}<br>
</body>
</html>
```

표현언어로 표현 단순화하기

❖ EL

- 조건문 활용

```
<% if(input==null) {%>  
    텅 빈 객체(null)입니다.  
<%}%>
```



```
<c:if test=${empty input}>  
    텅 빈 객체(null)입니다.  
</c:if>
```

표현언어로 요청 파라미터 처리하기

❖ 요청 파라미터 관련 EL 내장 객체

내장 객체	설명
param	JSP의 내장 객체인 request의 <code>getParameter()</code> 와 동일한 역할인 파라미터 값을 알려준다.
paramValues	동일한 이름으로 전달되는 파라미터 값들을 배열 형태로 얻어오는 데 사용하는 request의 <code>getParameterValues()</code> 와 동일한 역할을 한다.

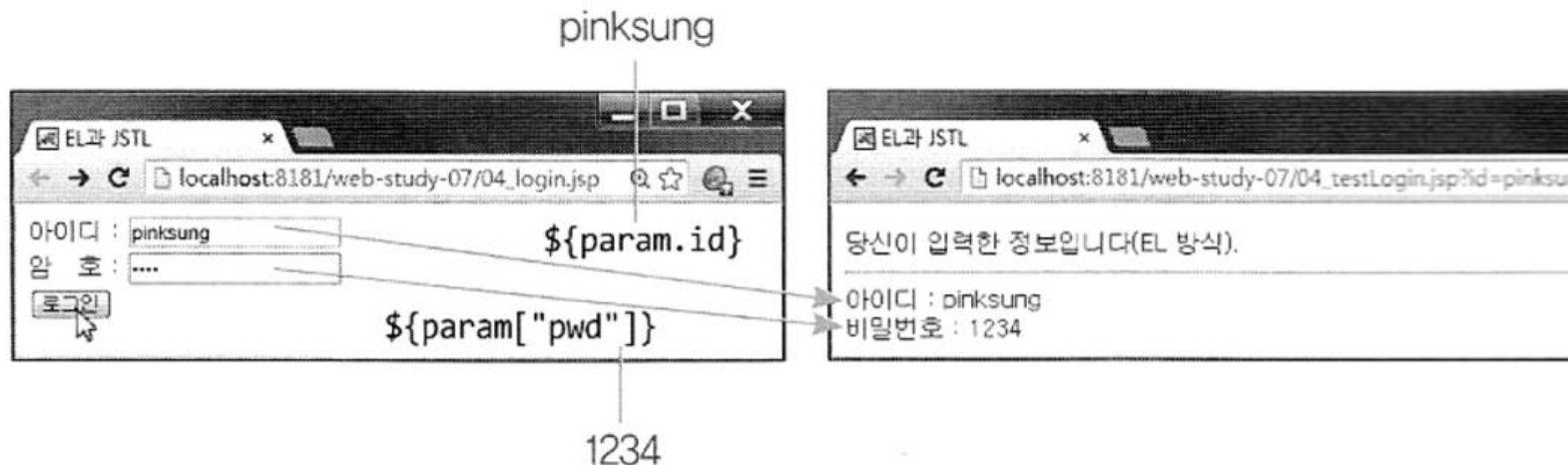
표현언어로 요청 파라미터 처리하기

❖ 요청 파라미터 관련 EL 내장 객체

- param
 - 요청 파라미터 값 추출

`${param.파라미터명}`
`${param[파라미터명]}`

→ 파라미터명을 변수로 가지고 있는 경우 사용



표현언어로 요청 파라미터 처리하기

❖ 요청 파라미터 관련 EL 내장 객체

- paramValues

```
<form action="04_research.jsp">
```

좋아하는 계절:

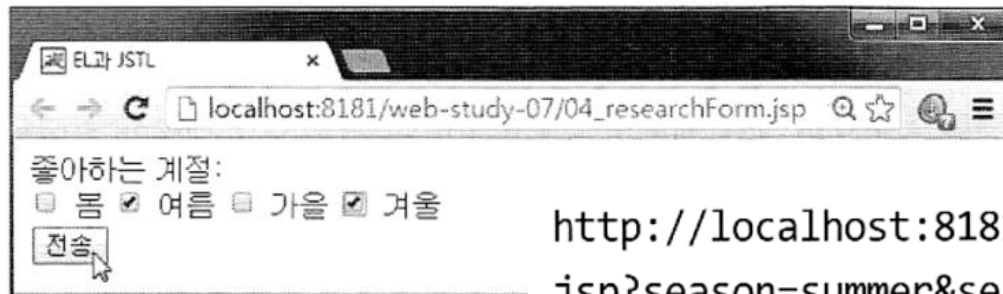
```
<input type="checkbox" name="season" value="spring"> 봄
```

```
<input type="checkbox" name="season" value="summer" > 여름
```

```
<input type="checkbox" name="season" value="fall"> 가을
```

```
<input type="checkbox" name="season" value="winter"> 겨울
```

```
</form>
```



http://localhost:8181/web-study-07/04_research.
jsp?season=summer&season=winter

표현언어로 요청 파라미터 처리하기

❖ 요청 파라미터 관련 EL 내장 객체

- paramValues

```
<%  
    String seasonArr[] = request.getParameterValues("season");  
    out.println("당신이 좋아하는 계절 : ");  
    for (String season : seasonArr) {  
        out.println(season+" ");  
    }  
%>
```



```
<c:forEach items="${paramValues.season}" var="season">  
    ${season}  
</c:forEach>
```

표현언어로 요청 파라미터 처리하기

❖ 예제

파일	설명
04_login.jsp	클라이언트가 ID와 암호를 입력할 수 있는 화면을 제공하기 위한 페이지이다.
04_testLogin.jsp	로그인 화면에서 제공한 정보를 가져와 출력하는 페이지이다.

표현언어로 요청 파라미터 처리하기

❖ 04_login.jsp

```
<body>
<form method="get" action="04_testLogin.jsp">
  <label for="userid"> 아이디 : </label>
  <input type="text" name="id" id="userid"><br>
  <label for="userpwd"> 암호 : </label>
  <input type="password" name="pwd" id="userpwd"><br>
  <input type="submit" value="로그인">
</form>
</body>
</html>
```

표현언어로 요청 파라미터 처리하기

❖ 04_loginTest.jsp

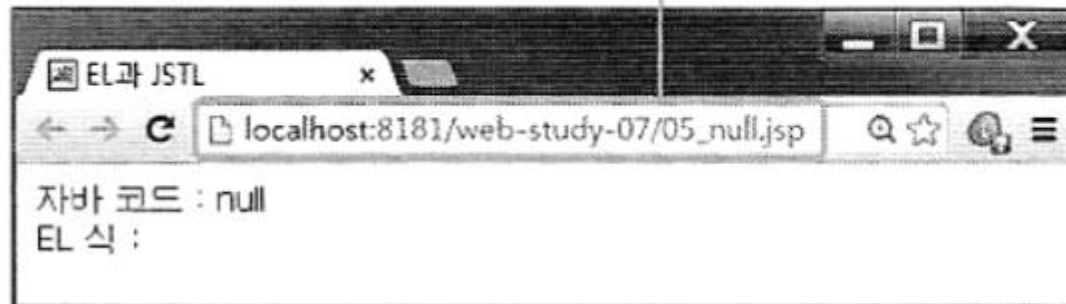
```
<body>
    당신이 입력한 정보입니다(고전적인 방식). <hr>
    아이디 : <%= request.getParameter("id")%> <br>
    비밀번호 : <%= request.getParameter("pwd")%> <br><br>
    당신이 입력한 정보입니다(EL 방식). <hr>
    아이디 : ${param.id} <br>
    비밀번호 : ${param["pwd"]}
</body>
</html>
```

표현언어로 요청 파라미터 처리하기

❖ 05_null.jsp

```
<body>  
    자바 코드 : <%=request.getParameter("id") %> <br>  
    EL 식 : ${param.id}  
</body>  
</html>
```

http://localhost:8181/web-study-07/05_null.jsp



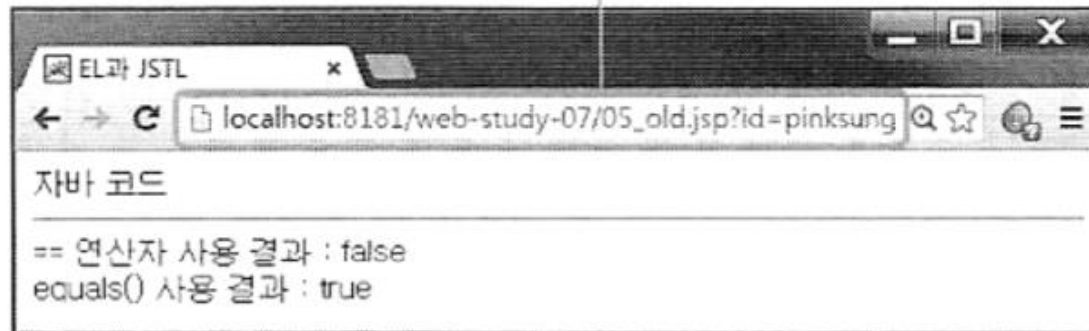
표현언어로 요청 파라미터 처리하기

❖ 05_old.jsp

```
<body>
  자바 코드 <br>
  == 연산자 사용 결과 :
    <%=request.getParameter("id")== "pinksung" %><br>

  equals() 사용 결과 :
    <%=request.getParameter("id").equals("pinksung") %><br>
</body>
</html>
```

http://localhost:8181/web-study-07/05_old.jsp?id=pinksung

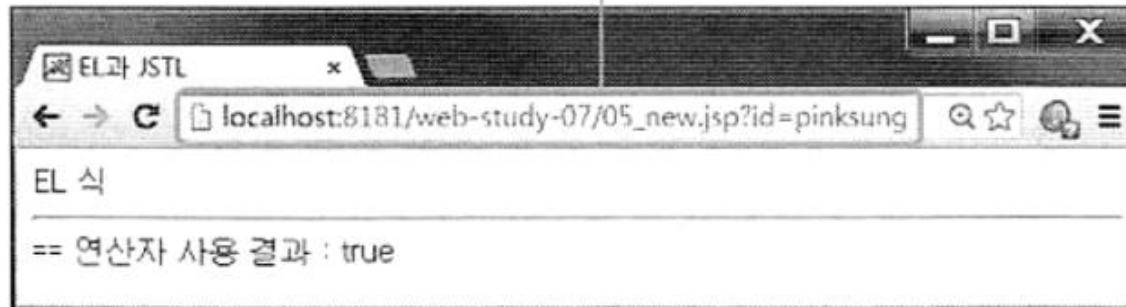


표현언어로 요청 파라미터 처리하기

❖ 05_new.jsp

```
<body>
    EL 식 <hr>
    == 연산자 사용 결과 : ${param.id=="pinksung"}<br>
</body>
</html>
```

http://localhost:8181/web-study-07/05_new.jsp?id=pinksung



표현언어로 요청 파라미터 처리하기

❖ 05_oldRenual.jsp

```
<body>
  자바 코드 <hr>
  <% if(request.getParameter("id") != null){ %>
    == 연산자 사용 결과 :
    <%=request.getParameter("id")== "pinksung" %><br>
    equals() 사용 결과 :

    <%=request.getParameter("id").equals("pinksung") %><br>
    <% }else{
      out.print("id를 전달해 주세요.");
    }
  %>
</body>
</html>
```

표현언어로 요청 파라미터 처리하기

❖ 06_addForm.jsp - EL의 자동 형변환

```
<form action="06_addition.jsp">  
    숫자 1 : <input type="text" name="num1"><br>  
    숫자 2 : <input type="text" name="num2"><br>  
    <input type="submit" value="전송">  
</form>  
</body>  
</html>
```

표현언어로 요청 파라미터 처리하기

❖ 06_addition - EL의 자동 형변환

```
고전적인 방식 : <!--스크립트릿에서의 복잡한 자바 코드 --%>
<%
String str1 = request.getParameter("num1");
String str2 = request.getParameter("num2");
int num1 = Integer.parseInt(str1);
int num2 = Integer.parseInt(str2);
%>
<%=num1%> + <%=num2%> = <%=num1 + num2%> <hr>
EL 방식 : <!-- 보다 깔끔하고 간단한 코드 --%>
    ${param.num1} + ${param.num2} = ${param.num1+param.num2}
</body>
</html>
```

표현 언어로 내장 객체 접근하기

❖ 영역 관련 EL 내장 객체

Category	내장 객체	설명
범위	pageScope	page 기본 객체에 저장된 속성의 〈속성, 값〉 매핑을 저장한 Map 객체
	requestScope	request 기본 객체에 저장된 속성의 〈속성, 값〉 매핑을 저장한 Map 객체
	sessionScope	session 기본 객체에 저장된 속성의 〈속성, 값〉 매핑을 저장한 Map 객체
	applicationScope	application 기본 객체에 저장된 속성의 〈속성, 값〉 매핑을 저장한 Map 객체

○ JSP내장 객체와 동일한 역할

- EL에서 속성명을 지정할 때 내장 객체를 지정하지 않으면
 - 작은 범위에서 큰 범위로 자동 검색
 - pageScope → requestScope → sessionScope → applicationScope

표현 언어로 내장 객체 접근하기

❖ 예제

- AdditionServlet.java 생성

```
package unit07;
:
@WebServlet("/AdditionServlet")
public class AdditionServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public AdditionServlet() {
        super();
    }

    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response)
        throws ServletException, IOException
    {
        doGet(request, response);
    }
}
```

표현 언어로 내장 객체 접근하기

❖ 예제

- AdditionServlet.java 생성

```
protected void doGet(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    int num1 = 20;
    int num2 = 10;
    int add = num1 + num2;
    request.setAttribute("num1", num1);
    request.setAttribute("num2", num2);
    request.setAttribute("add", add);

    RequestDispatcher dispatcher =
        request.getRequestDispatcher("07_addition.jsp");
    dispatcher.forward(request, response);
}
```

표현 언어로 내장 객체 접근하기

❖ 07_addition.jsp

```
<body>
  고전적인 방식 :
  <%
    int num1 = (Integer) request.getAttribute("num1");
    int num2 = (Integer) request.getAttribute("num2");
  %>
  <%=num1%> + <%=num2%> = <%=num1 + num2%><hr>

  EL 방식 : ${num1} + ${num2} = ${add}
</body>
</html>
```


표현 언어로 내장 객체 접근하기

❖ 영역 관련 EL 내장 객체

속성	JSP 내장 객체	서블릿 클래스	표현 언어의 내장 객체
page 속성	pageContext	javax.servlet.jsp.JspContext 클래스	pageScope
request 속성	request	javax.servlet.ServletRequest 인터페이스	requestScope
session 속성	session	javax.servlet.http.HttpSession 인터페이스	sessionScope
application 속성	application	javax.servlet.ServletContext 인터페이스	applicationScope

표현 언어로 내장 객체 접근하기

❖ 영역 관련 EL 내장 객체

■ 예 : 자바 코드

```
pageContext.getAttribute("num1");
```

■ 예 : 자바 코드

```
request.getAttribute("num1");
```

■ 예 : 자바 코드

```
session.getAttribute("num1");
```

■ 예 : 자바 코드

```
application.getAttribute("num1");
```

■ 예 : 표현 언어

```
${pageScope.num1}
```

■ 예 : 표현 언어

```
${requestScope.num1}
```

■ 예 : 표현 언어

```
${sessionScope.num1}
```

■ 예 : 표현 언어

```
${applicationScope.num1}
```

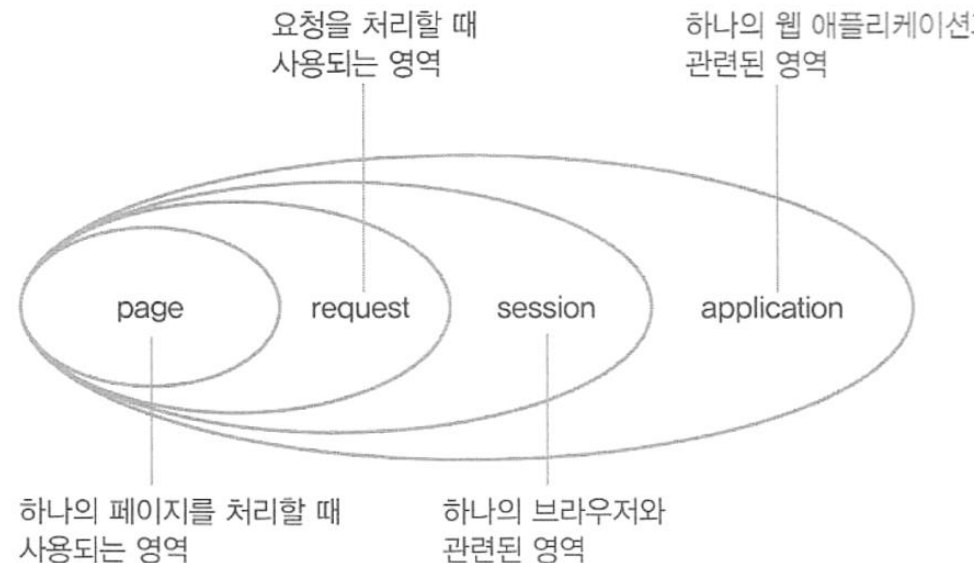
표현 언어로 내장 객체 접근하기

❖ 영역 관련 EL 내장 객체

- EL에서 속성명을 지정할 때 내장 객체를 지정하지 않으면
 - 작은 범위에서 큰 범위로 자동 검색
 - pageScope → requestScope → sessionScope → applicationScope

```
pageContext.setAttribute("name", "page man");  
request.setAttribute("name", "request man");  
session.setAttribute("name", "session man");  
application.setAttribute("name", "application man");
```

`${name}` → ?



표현 언어로 내장 객체 접근하기

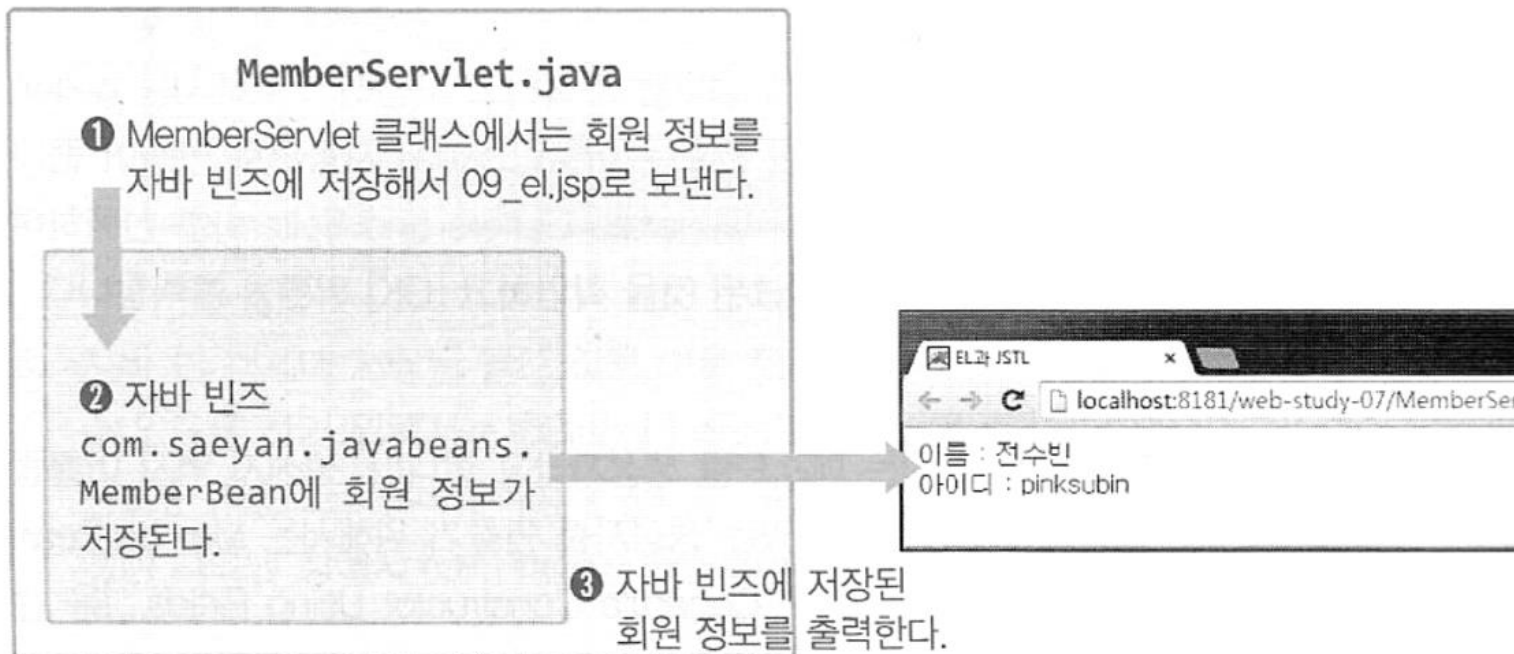
❖ 08_scope.jsp

```
<%
    pageContext.setAttribute("name", "page man");
    request.setAttribute("name", "request man");
    session.setAttribute("name", "session man");
    application.setAttribute("name", "application man");
%>
name : ${name} <hr>
page 속성 : ${pageScope.name} <br>
request 속성 : ${requestScope.name} <br>
session 속성 : ${sessionScope.name} <br>
application 속성 : ${applicationScope.name} <br>
```

표현 언어로 내장 객체 접근하기

❖ 6장 리뉴얼

파일 이름	설명
MemberBean.java	회원의 정보를 저장하는 자바 빈입니다.
MemberServlet.java	회원 정보를 저장한 자바 빈 객체를 생성하기 위한 서블릿 클래스입니다.
09_el.jsp	MemberServlet에서 생성한 자바 빈에 저장된 회원 정보를 얻어와 출력합니다.



표현 언어로 내장 객체 접근하기

❖ MemberBean.java

```
public class MemberBean {  
    private String name;  
    private String userid;  
  
    // 생성자  
    // Getter/Setter  
    // toString()  
}
```

표현 언어로 내장 객체 접근하기

❖ MemberServlet.java

```
import com.saeyan.javabeans.MemberBean;

@WebServlet("/MemberServlet")
public class MemberServlet extends HttpServlet {
    :

    protected void doGet(HttpServletRequest request,
                        HttpServletResponse response)
                        throws ServletException, IOException {
        MemberBean member=new MemberBean("전수빈", "pinksubin");
        request.setAttribute("member", member);
        RequestDispatcher dispatcher =
            request.getRequestDispatcher("09_e1.jsp");
        dispatcher.forward(request, response);
    }
}
```

표현 언어로 내장 객체 접근하기

❖ 09_el.jsp

```
<body>
  이름 : ${member.name}<br>
  아이디 : ${member["userid"]}
</body>
</html>
```