



JSP (EL & JSTL)

ThS. Nguyễn Nghiệm
0913.745.789 - NghiemN@fpt.edu.vn



● EL

- ✱ Truy xuất attribute
- ✱ Truy xuất thuộc tính bean
- ✱ Truy xuất Collection
- ✱ Truy xuất Map
- ✱ Truy xuất Parameter, cookie và header


● JSTL

- ✱ Thư viện thẻ core
 - <c:set>, <c:remove>
 - <c:if>, <c:choose>, <c:forEach>
- ✱ Thư viện thẻ định dạng
 - <fmt:formatNumber>
 - <fmt:formatDate>
- ✱ Thư viện hàm





EXPRESSION LANGUAGE

- EL Được sử dụng để đơn giản hóa việc truy xuất các thành phần và xuất kết quả trong lập trình JSP 2.0.
- Cú pháp:
 - ✱ `${<biểuthức>}`
- Ví dụ:
 - ✱ `${xyz}` , `${sessionScope['xyz']}`, `${param.xyz}`...
- Trong <biểu thức> thường chứa các thành phần sau
 - ✱ Attribute: `${message}` 
 - ✱ Parameter: `${param['fullname']}`
 - ✱ Cookie: `${cookie['view'].value}`

`{attribute}`



TRUY CẬP ATTRIBUTE

- Trong JSP có 4 phạm vi chia sẻ dữ liệu
 - ✱ Request: **requestScope**
 - ✱ Session: **sessionScope**
 - ✱ Application: **applicationScope**
 - ✱ Page: **pageScope**
- Làm việc với các attribute trong java.
 - ✱ `Double x = (Double)session.getAttribute("salary");`
 - ✱ `Double netSalary= x*1.8;`
 - ✱ `out.print(netSalary);`
- Làm việc với các attribute với EL
 - ✱ `${sessionScope['salary']*1.8}`



TRUY XUẤT ATTRIBUTE

● Truy xuất theo Scope

- ✱ `${ pageScope['x']}` hoặc `${ pageScope.x}`
- ✱ `${ requestScope['x']}` hoặc `${ requestScope.x}`
- ✱ `${ sessionScope['x']}` hoặc `${ sessionScope.x}`
- ✱ `${ applicationScope['x']}` hoặc `${ applicationScope.x}`

● Truy tìm Attribute trong tất cả Scope

- ✱ `${message}`
- ✱ Attribute message được truy tìm theo trình tự:
 - page->request->session->application
- ✱ Nếu tìm thấy thì dừng lại, ngược lại cho giá trị rỗng



TRUY XUẤT THUỘC TÍNH CỦA BEAN

- Lớp **JavaBean** là lớp
 - ✱ Có Constructor mặc định
 - ✱ Có getter và setter để đọc ghi dữ liệu
- Cú pháp truy xuất:
 - ✱ `${bean.property} -> cart.getProperty()`
- Ví dụ:
 - ✱ `${cart.count} -> cart.getCount()`
 - ✱ `${mail.from} -> mail.getFrom()`



TRUY XUẤT MẢNG VÀ TẬP HỢP

- Sử dụng chỉ số để truy xuất các phần tử trong tập hợp và mảng.

Servlet

```
List<String> names = new List<String>();  
names.add("Nguyễn Văn Tèo");  
names.add("Phạm Hữu Khương")  
request.setAttribute("items", names);
```

JSP

```
${items[0]}  
${items[1]}
```



TRUY XUẤT MAP

- Sử dụng key để phân biệt các phần tử muốn truy xuất trong map.

Servlet

```
Map<String, String> m = new HashMap<String, String>();  
m.put("name", "Nguyen Nghiem");  
m.put("birthday", new java.util.Date());  
request.setAttribute("items", m);
```

JSP

```
${items['name']}  
${items.birthday}
```




TRUY XUẤT PARAMETER, COOKIE

- Truy xuất tham số đơn giản

- ✱ `${param.['name']}`

- ✱ `${param.name}`

- Truy xuất cookie

- ✱ `${cookie['user'].value}`

- ✱ `${cookie.user.value}`



JSTL-JAVA STANDARD TAG LIBRARY

Edited by Foxit Reader
Copyright(C) by Foxit Software Company,2005-2007
For Evaluation Only.

- Java cung cấp nhiều bộ thư viện thẻ chuẩn hỗ trợ lập trình JSP với nhiều mục đích khác nhau.



- Các thư viện sẽ sử dụng trong bài học

- ✱ Thư viện cơ bản

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

- ✱ Thư viện định dạng

`<%@ taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt" %>`

- ✱ Thư viện hàm

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



THƯ VIỆN CORE

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

- **<c:if>**

- ✱ Tương tự lệnh if

- **<c:choose>**

- ✱ Tương tự if...else if...else

- **<c:forEach>**

- ✱ Tương tự for-each

- **<c:set>**

- ✱ Tương tự: scope.setAttribute()

- **<c:remove>**

- ✱ Tương tự scope.removeAttribute()



THẺ <C:IF>

- Ví dụ sau sẽ hiển thị liên kết Delete khi giá trị của thuộc tính role của user là 'admin'

```
<c:if test="{sessionScope['user'].role == 'admin'}">  
    <a href='delete.do'>Delete</a>  
</c:if>
```



THE <C:CHOOSE>

```
<c:choose>
  <c:when test="{product.latest == true}">
    
  </c:when>
  <c:when test="{product.discount == true}">
    
  </c:when>
  <c:otherwise>
    
  </c:otherwise>
</c:choose>
```



<C:FOREACH>

```
<c:forEach begin="1" end="6" varStatus="status">  
    <h${status.index}>Hello World</h${status.index}>  
</c:forEach>
```

```
<c:forEach var="p" items="${products}" begin="10" end="25" varStatus="status">  
    <li>Vị trí: ${status.index}</li>  
    <li>Tên hàng hóa: ${p.name}</li>  
</c:forEach>
```





<C:SET> & <C:REMOVE>

● <c:set>

- ✱ Dùng để thiết lập attribute hoặc thuộc tính của bean.
- ✱ `<c:set var="name" value="value" scope="session"/>`
- ✱ `<c:set var="name" scope="session">value</c:set>`
- ✱ Tương tự `session.setAttribute("name", "value")`

● <c:remove>

- ✱ Dùng để xóa một attribute
- ✱ `<c:remove var="name" scope="session"/>`
- ✱ Tương đương `session.removeAttribute("name")`



ĐỊNH DẠNG

`<%@ taglib uri="http://java.sun.com/jstl/fmt_rt" prefix="fmt"%>`

- `<fmt:formatNumber>`

- ✱ Định dạng số

- `<fmt:formatDate>`

- ✱ Định dạng ngày

- Ví dụ

- Định dạng tiền tệ và%

- ✱ `<fmt:formatNumber value="1000000" type="currency" />`

- ✱ `<fmt:formatNumber value="0.51" type="percent" />`

- Định dạng ngày

- ✱ `<fmt:formatDate value="${date}" pattern="dd-MM-yyyy" />`



THƯ VIỆN HÀM

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>

Tên hàm	Đối số	Trả về	Mô tả mục đích
fn:contains	String, String	boolean	Chuỗi (1) có chứa chuỗi (2) hay không
fn:containsIgnoreCase	String, String	boolean	Chuỗi (1) có chứa chuỗi (2) hay không (không phân biệt hoa thường)
fn:endsWith	String, String	boolean	Chuỗi (1) có kết thúc bởi (2) hay không
fn:escapeXML	String	String	Mã hóa thành thực thể các ký tự phạm cú pháp XML
fn:indexOf	String, String	int	Tìm vị trí xuất hiện đầu tiên của chuỗi (2) trong chuỗi (1)
fn:join	String[], String	String	Gia nhập các phần tử trong mảng (1) thành chuỗi sử dụng chuỗi(2) như là chuỗi phân cách.
fn:length	Map; array; Collection; Iterator; Enumeration; or String	int	Tìm độ dài của chuỗi hay số lượng các phần tử trong tập hợp.



THƯ VIỆN HÀM

Tên hàm	Đối số	Trả về	Mô tả mục đích
fn:replace	String, String, String	String	Thay thế chuỗi (1) bởi chuỗi (3) trong chuỗi (1)
fn:split	String, String	String[]	Tách chuỗi (1) thành mảng sử dụng chuỗi (2) như chuỗi phân cách
fn:startsWith	String, String	boolean	Chuỗi đối số thứ nhất có bắt đầu bởi chuỗi đối số thứ hai hay không
fn:substring	String, int, int	String	Lấy chuỗi trong chuỗi (1) tính từ vị trí (1) cho đến vị trí (3)
fn:substringAfter	String, String	String	Lấy chuỗi con trong chuỗi (1) đứng sau chuỗi (2)
fn:substringBefore	String, String	String	Lấy chuỗi con trong chuỗi (1) đứng trước chuỗi (2)
fn:toLowerCase	String	String	Đổi chuỗi sang chữ thường
fn:toUpperCase	String	String	Đổi chuỗi sang chữ HOA
fn:trim	String	String	Cắt bỏ khoảng trắng 2 đầu chuỗi



● EL

- ✱ Truy xuất attribute
- ✱ Truy xuất thuộc tính bean
- ✱ Truy xuất Collection
- ✱ Truy xuất Map
- ✱ Truy xuất Parameter, cookie và header

● JSTL

- ✱ Thư viện thẻ core
 - <c:set>, <c:remove>
 - <c:if>, <c:choose>, <c:forEach>
- ✱ Thư viện thẻ định dạng
 - <fmt:formatNumber>
 - <fmt:formatDate>
- ✱ Thư viện hàm