CS3220 Web and Internet Programming

JSP Standard Tag Library (JSTL)

Chengyu Sun
California State University, Los Angeles

Custom Tags in ASP.NET

```
<p>Enter a number from 1 to 100:

<asp:TextBox id="tbox1" runat="server" />
<asp:Button Text="Submit" runat="server" />

</p>
```

Custom Tags in ColdFusion

```
<cfform action = "cftextinput.cfm" method="post">

    <cftextinput name = "myInput"
        message = "This field must not be blank"
        required = "Yes" />

    <input type = "Submit" value ="submit" />

</cfform>
```

Simple JSTL Example

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html><head><title>JSTL Hello</title></head>
<body>

<c:out value="Hello World in JSTL." />

</body>
</html>
```

taglib Directive

- URI
  - A unique identifier for the tag library
  - NOT a *URL*
- Prefix
  - A short name for the tag library
  - Could be an arbitrary name

JSP Standard Tag Library (JSTL)

Library	URI	Prefix
Core	http://java.sun.com/jsp/jstl/core	c
XML Processing	http://java.sun.com/jsp/jstl/xml	x
I18N Formatting	http://java.sun.com/jsp/jstl/fmt	fmt
Database Access	http://java.sun.com/jsp/jstl/sql	sql
Functions	http://java.sun.com/jsp/jstl/functions	fn

http://download.oracle.com/docs/cd/E17802_01/products/products/jsp/jstl/1.1/docs/tlddocs/index.html

Important JSTL Tags and Functions

- Flow control
  - <c:if>
  - <c:choose>
    - <c:when>
    - <c:otherwise>
  - <c:forEach>
- URL
  - <c:url>
- Output
  - <c:out>
- Formatting
  - <fmt:formatDate>
  - <fmt:formatNumber>
- Functions
  - fn:length()

Branch Tags …

<c:if test="**${**a.weekdays[0] == 'mon'**}**">Monday</c:if>

The *type* of the `test` attribute is `boolean`, which requires EL; "a.weekdays[0]=='mon'" would be just a String.

… Branch Tags

```
<c:choose>
    <c:when test="${a.weekdays[3] == 'tue'}">
        Tuesday
    </c:when>
    <c:when test="${a.weekdays[3] == 'wed'}">
        Wednesday</c:when>
    <c:when test="${a.weekdays[3] == 'thr'}">
        Thursday
    </c:when>
    <c:otherwise>Friday</c:otherwise>
</c:choose>
```

Loop Tag <c:forEach> …

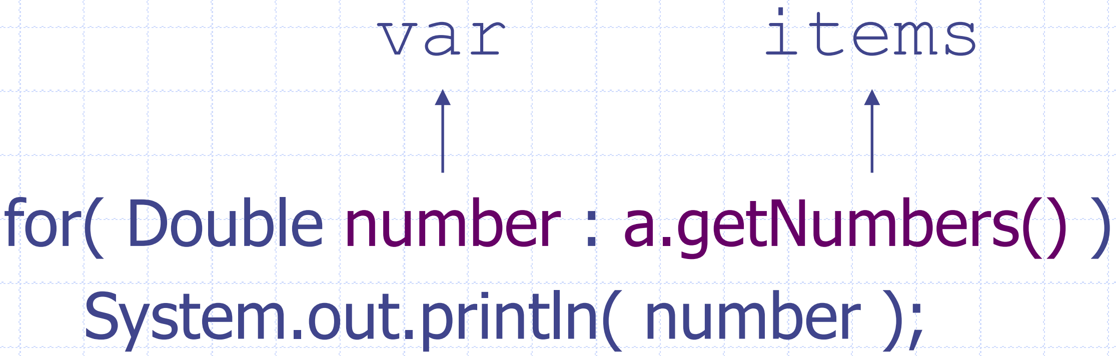
The *type* of the `items` attribute is *Collection* (i.e. array, list, set, etc.)

```
<ul>
    <c:forEach items="${a.numbers}" var="number">
        <li>${number}</li>
    </c:forEach>
</ul>
```

The *type* of the `var` attribute is simply `String`

… Loop Tag <c:forEach>

- It's similar to the "enhanced for loop" in Java



var

items

```
for( Double number : a.getNumbers() )
    System.out.println( number );
```

Another Way to Use <c:forEach>

```
<ul>
    <c:forEach begin="0" end="10" step="1" var="i">
        <li>${a.numbers[i]}</li>
    </c:forEach>
</ul>
```

- Just like a regular for-loop in Java

varStatus

- Use `varStatus` variable to get the information of each iteration
- http://docs.oracle.com/javaee/5/jstl/1.1/docs/api/javax/servlet/jsp/jstl/core/LoopTagStatus.html

```
<ul>
    <c:forEach items="${a.numbers}" var="number"
          varStatus="status">
        <li>${status.index}: ${number}</li>
    </c:forEach>
</ul>
```

URL Tag <c:url>

```
<a href="<c:url value='/JSTLTest' />">A</a>
```

```
<c:url value="/JSTLTest" />
```

For <c:url>, a URL starts with "/" is relative to the application root instead of the server root.

Output Tag <c:out>

<c:out value="abc" /> → "abc"

<c:out value="${a.id}" /> → ${a.id}

<c:out value="a<b>c" /> → ??

- You want to use <c:out> if
  - *value* contains special characters
  - *value* is a Java.io.Reader object

Some Special Characters

<	&lt;
>	&gt;
&	&amp;
'	&#039;
"	&#034;

Format Date and Time with <fmt:formatDate>

```
<fmt:formatDate value="${a.date}" type="date" />

<fmt:formatDate value="${a.date}" type="time" />

<fmt:formatDate value="${a.date}" type="both" />

<fmt:formatDate value="${a.date}"
        pattern="yyyy-M-d h:m:s a" />
```

See http://download.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html for the date formatting patterns.

Format Numbers with <fmt:formatNumber>

```
<fmt:formatNumber value="${number}" type="number" />

<fmt:formatNumber value="${number}" type="currency" />

<fmt:formatNumber value="${number}" type="percent" />

<fmt:formatNumber value="${number}" pattern="0.##" />
```

See https://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html for the number formatting patterns.

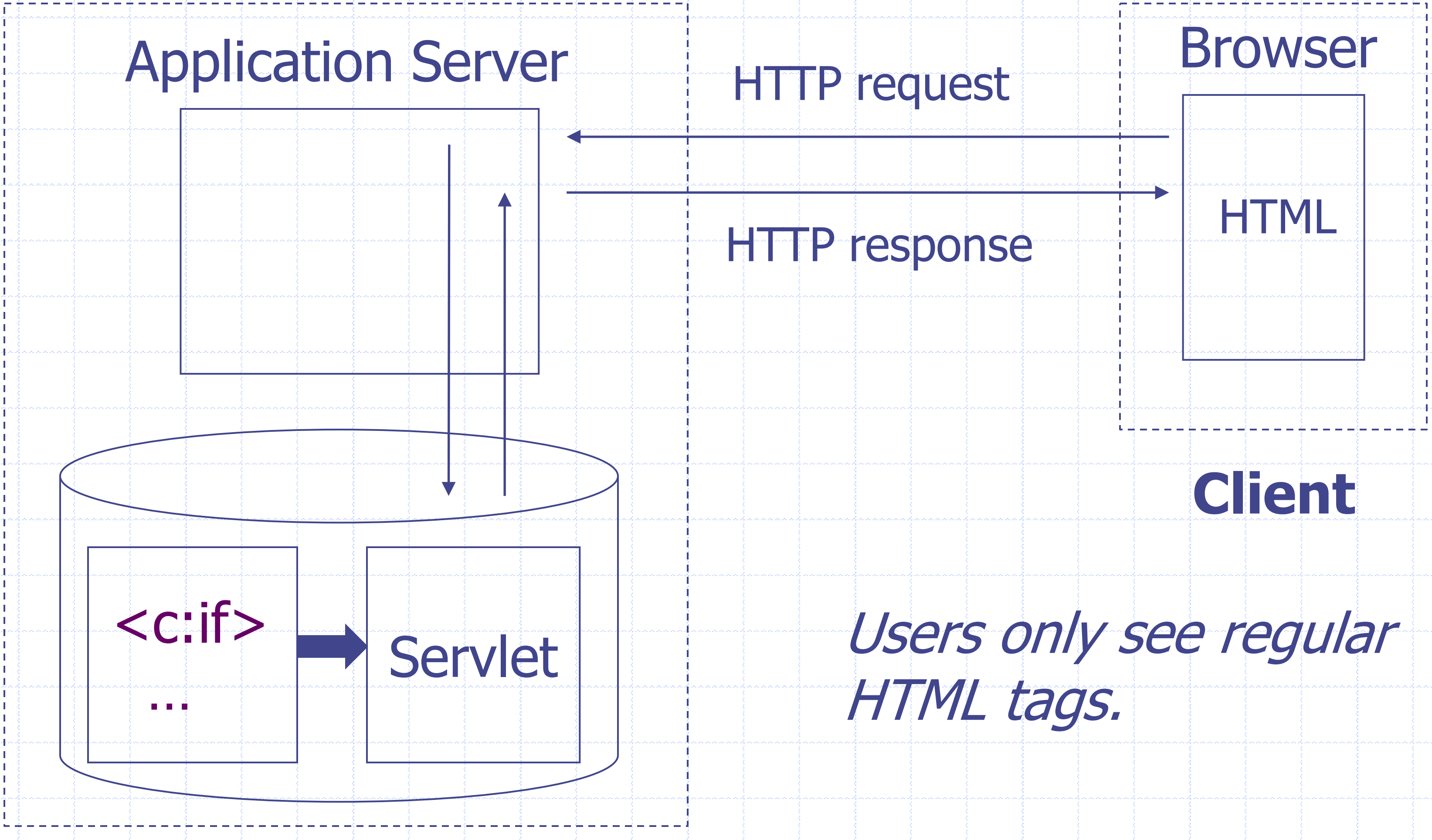
Use JSTL function fn:length()

${ fn:length("cs3220") **}** → Length of a String

${ fn:length(a.numbers) **}** → Size of a collection

- Can only be used in EL
- More at http://docs.oracle.com/cd/E17802_01/products/products/jsp/jstl/1.1/docs/tlddocs/index.html

Custom Tags Are "Server Tags"



Users only see regular HTML tags.

Server

JSP As A Template Engine

- Expression Language (EL)
  - Access to objects and properties
  - Simple expressions
- JSP Standard Tag Library (JSTL)
  - Control flow structures (i.e. branch and loop)
  - Additional utilities such as formatting
- Example: convert GuestBook (Servlet) to MVC