JSP 2.0 XML Cheat Sheet

➤ General

- `<%@    <jsp:directive.[directiveName]...`
- `<%!    <jsp:declaration> int i=0; </jsp:declaration>`
- `<%=    <jsp:expression> myVar.getProp() </jsp:expression>`
- `<%     <jsp:scriptlet> if (... </jsp:scriptlet>`

scope is `page|request|session|application`

```
<?xml version="1.0" ?>
   <jsp:root version="1.2|2.0"
   [ xmlns:taglibPrefix="URI" ]+ ...
   xmlns     = "http://www.w3c.org/1999/xhtml"
   xmlns:jsp = "http://java.sun.com/JSP/Page"
   xmlns:c   = "http://java.sun.com/jstl/core"
   xmlns:x   = "http://java.sun.com/jstl/xml"
   ...
```

➤ JSTL

```
xmlns:c="http://java.sun.com/jsp/jstl/core"
```

```
<c:url value="" var="var" scope="SCOPE" ... <c:param
<c:out value=" " escapeXml="true"
```

```
<c:if test=""
<c:choose
   <c:when test=""
   <c:otherwise
<c:forEach var="" varStatus="" items="" begin="startIndex" end="" step=""
```

```
<c:forTokens  var="" varStatus="" items="" delims="" begin="startIndex" end="" step=""
```

```
<c:catch var=""
```

```
<c:set value="or element body" var="" scope="SCOPE" target="" property=""
<c:remove var="" scope=""
```

```
<c:import url="" var="" scope="SCOPE" charEncoding="" ...
   <c:param name="" value=""
<c:redirect   ...
   <c:param name="" value=""
```

```
xmlns:fmt="http://java.sun.com/jsp/jstl/fmt"
<fmt:requestEncoding value="encoding name"
```

```
<fmt:message ... <fmt:param ... TBD
<fmt:bundle basename="" prefix=""
<fmt:setBundle  basename=""
   var="" scope="SCOPE" value=""
<fmt:formatNumber ... TBD
```

```
<fmt:formatDate
   var="" scope="SCOPE" value=""
   pattern="see java.text.SimpleDateFormat" type="date|time|both"
   dateStyle="default|short|medium|long|full"
   timeStyle="default|short|medium|long|full"
   timeZone=""
```

```
<fmt:timeZone value=""
<fmt:setTimeZone value=""
   var="" scope="SCOPE" value=""
```

```
<fmt:setLocale value=""
   var="" scope="SCOPE" value=""
```

```
<fmt:parseDate
```

```
<fmt:parseNumber
```

Tag or JSP Files

```
<jsp:directive.include
   file="a page relative or context relative URI path for the file to include"
```

```
<jsp:include page="page- or context-relative URI" flush="false"
   <jsp:param name="" value=""
```

```
<jsp:element name=""
   <jsp:attribute name="" trim="true" (body is value)
   <jsp:body (required when <jsp:attribute is used)
   (It's not possible have conditional attributes using these tags.)
```

```
<jsp:output
   omit-xml-declaration="true|yes|false|no"
   doctype-root-element=""
   doctype-public="publicID" doctype-system="systemID"
```

```
<jsp:getProperty name="bean variable name" property=""
```

```
<jsp:setProperty name="bean variable name" property="*|names"
   param="name of request parameter" value=""
```

```
<jsp:useBean id="" class="" beanName="" type="" scope="SCOPE"
```

`<jsp:text` preserves whitespace

```
<jsp:plugin ... <jsp:fallback
   <jsp:params
        <jsp:param name="" value=""
```

Tag Files Only

```
<jsp:directive.tag  description="" example=""
   body-content="empty|scriptless|tagdependent"
   dynamic-attributes="the name of the variable to hold dynamic attributes"
   import=""
   pageEncoding="ISO-8859-1"
```

```
<jsp:directive.attribute name=""
   description=""
   required="true|false"
   fragment="true|false"
   rtexprvalue="true|false"
   type="String classes and interfaces (no primitive types)"
```

```
<jsp:directive.variable
   alias="name of the local page scope variable the tag file uses to hold the value"
   name-from-attribute="page-scope variable name receiving value"
```

```
    declare="true (false to prevent export of variable)"
    description=""
    name-given="name of the invoking page's variable used to export the value"
    scope="NESTED|AT_BEGIN|AT_END"
    variable-class="class or interface"
```

```
<jsp:doBody var="" varReader="" scope="SCOPE"
```

```
<jsp:invoke fragment="attribute that defines the fragment"
    scope="SCOPE" var="" varReader=""
```

Page Only

```
<jsp:directive.page autoFlush="true" buffer="8kb"
    contentType="mimeType and encoding UTF-8, eg. 'text/html; charset=UTF-8'"
    errorPage="page or context relative path"
    extends="class name"
    import="package list"
    isErrorPage="false" isThreadSafe="true"
    pageEncoding="encoding of JSP page file and default contentType encoding"
    session="true"
```

```
<jsp:forward page="page or context relative path".
    <jsp:param name="" value=""
```

Scriptlets

pageContext	findAttribute, getAttribute, getAttributesScope, getAttributeNamesInScope, setAttribute	Page	javax.servlet.jsp.PageContext
out	clear, clearBuffer, flush, getBufferSize, getRemaining	Page	javax.servlet.jsp.JspWriter
config	getInitParameter, getInitParameterNames	Page	javax.servlet.ServletConfig
exception	getMessage, getLocalizedMessage, printStackTrace, toString	Page	java.lang.Throwable
page		Page	java.lang.Object
response		Page	javax.servlet.ServletResponse
request	getAttribute, getParameter, getParameterNames, getParameterValues, setAttribute	Request	javax.servlet.ServletRequest
session	getAttribute, getId, setAttribute	Session	javax.servlet.http.HttpSession
application	getAttribute, getMimeType, getRealPath, setAttribute	Application	javax.servlet.ServletContext

EL

operations and | &amp;&amp; | or | || | '+' | '-' | * | / | div | % | mod | gt | lt | ge | le | ==| eq | ne {'-'| ! | not | empty}

pageContext The context for the JSP page. Provides access to various objects, including servletContext, session, request, and response.

pageScope Maps page-scoped variable names to their values.

requestScope Maps request-scoped variable names to their values.

sessionScope Maps session-scoped variable names to their values.

applicationScopeMaps application-scoped variable names to their values.

param Maps a request parameter to a single String parameter value (obtained by calling ServletReqwuest.getParameter(String name)).

paramValues Maps a request parameter name to an array of String values for that parameter name (obtained by calling ServletRequest.getParameterValues(String name)).

`header` Maps a request header name to a single String header value (obtained by calling ServletRequest.getHeader(String name)).

`headerValues` Maps a request header name to an array of String values for that header (obtained by calling ServletRequest.getHeaders(String)).

`cookie` Maps a cookie name to a single Cookie object. Cookies are retrieved according to the semantics of HttpServletRequest.getCookies(). If same name is shared by multiple cookies, an implementation must use the first encountered in the array of Cookie objects returned by the getCookies() method. However, the ordering of cookies currently unsspecified in the Servlet specification.

`initParam` Maps a context initialization parameter name to a String parameter value (obtained by calling ServletContext.getInitparameter(String name)).

- `boolean  fn:contains(string, substring)`
- `boolean  fn:containsIgnoreCase(string, substring)`
- `boolean  fn:startsWith(string, prefix)`
- `boolean  fn:endsWith(string, suffix)`
- `String[] fn:split(string, separator)`
- `String   fn:substring(string, begin, end)`
- `String   fn:substringBefore(string, substring)`
- `String   fn:substringAfter(string, substring)`
- `String   fn:replace(string, before, after)`
- `String   fn:toLowerCase(string)`
- `String   fn:toUpperCase(string)`
- `String   fn:trim(string)`
- `int      fn:indexOf(string, suffix)`
- `int      fn:length(String or Collection)`
- `String   fn:join(String[], separator)`
- `String   fn:escapeXml(string)`