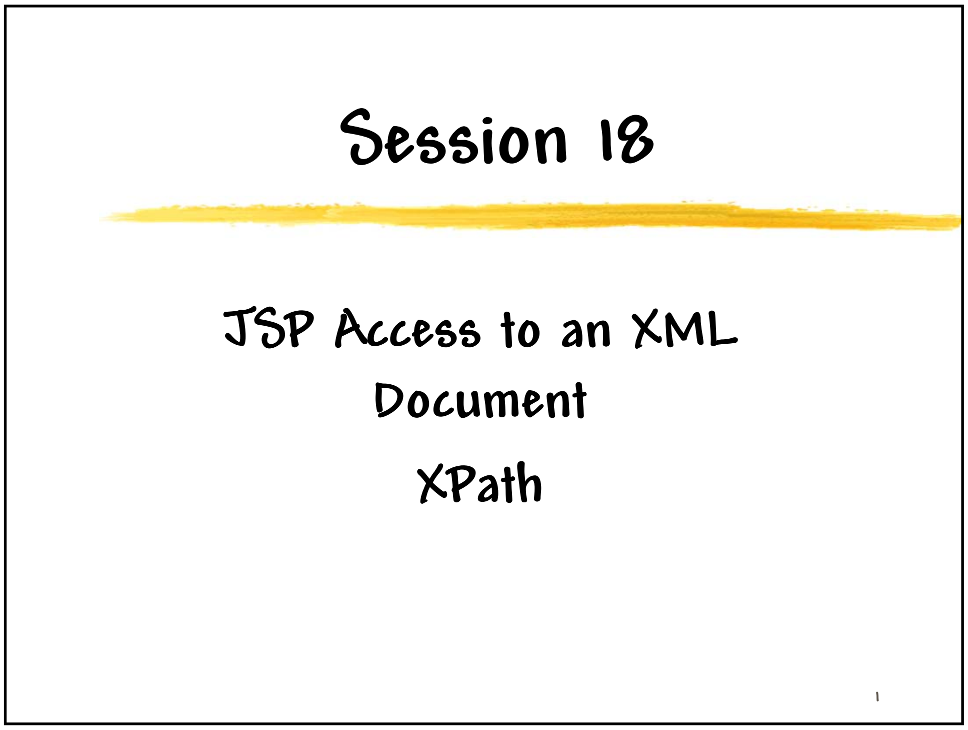
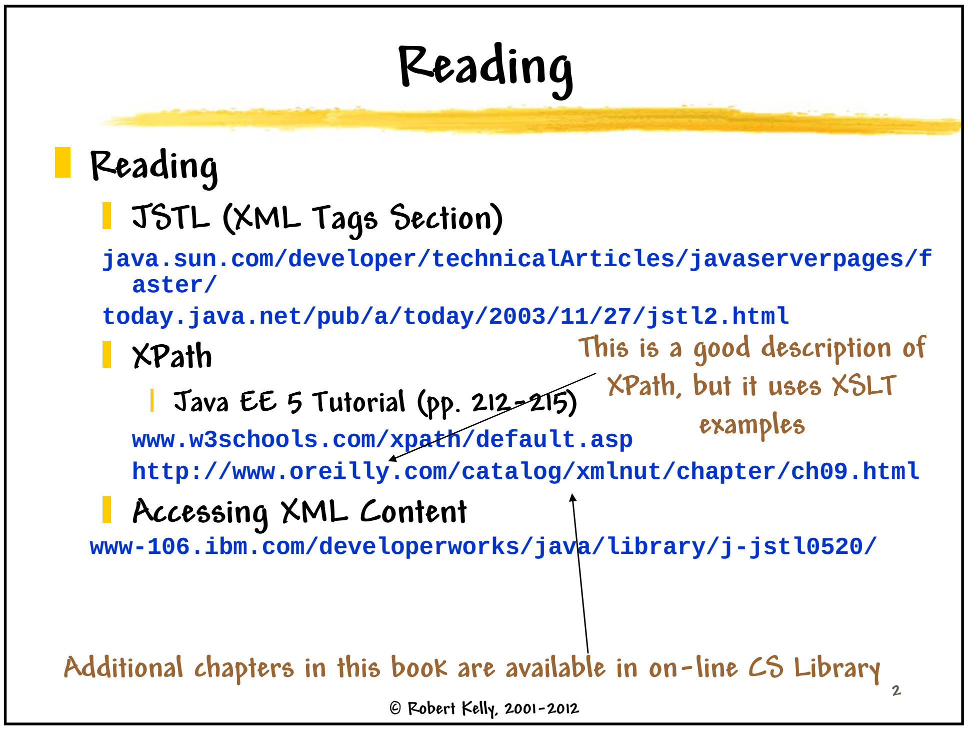


Session 18

JSP Access to an XML Document

XPath

1



Reading

- Reading
  - JSTL (XML Tags Section)

    java.sun.com/developer/technicalArticles/javaserverpages/faster/

    today.java.net/pub/a/today/2003/11/27/jstl2.html
  - XPath
    - Java EE 5 Tutorial (pp. 212 - 215)

    www.w3schools.com/xpath/default.asp

    http://www.oreilly.com/catalog/xmlnut/chapter/ch09.html
  - Accessing XML Content

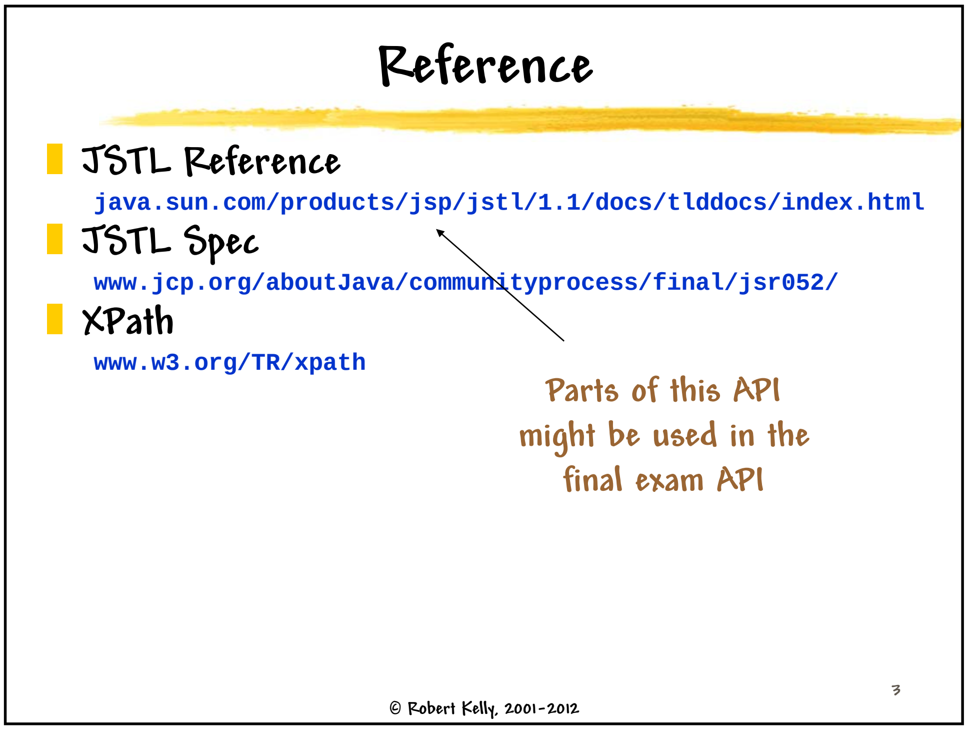
    www-106.ibm.com/developerworks/java/library/j-jstl0520/

This is a good description of XPath, but it uses XSLT examples

Additional chapters in this book are available in on-line CS Library

2

© Robert Kelly, 2001-2012



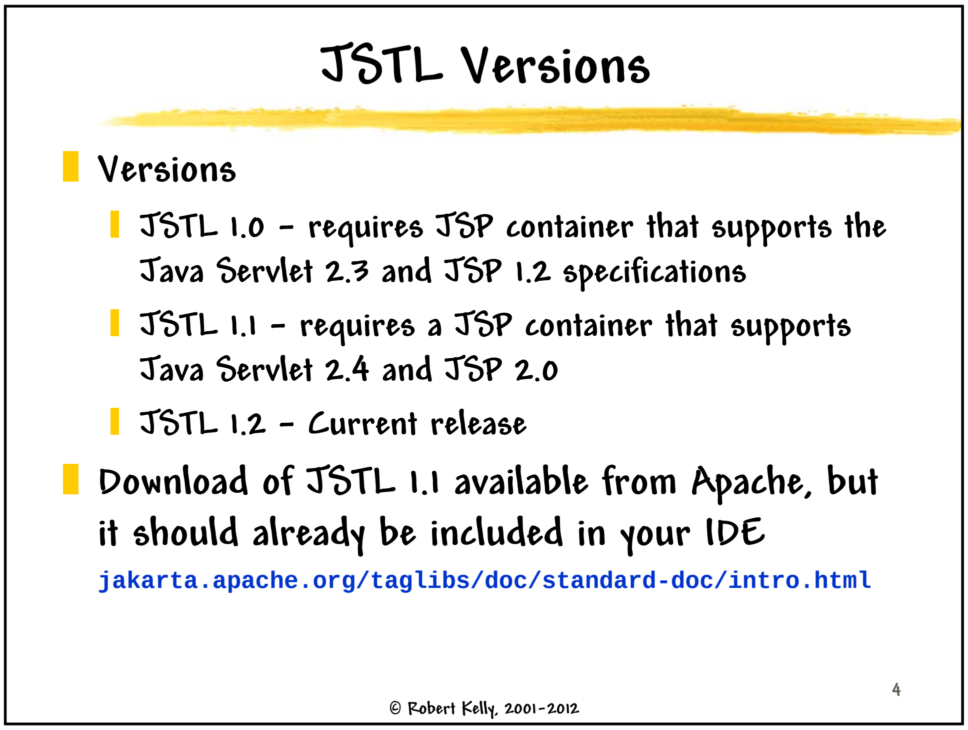
Reference

- JSTL Reference
  java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html
- JSTL Spec
  www.jcp.org/aboutJava/communityprocess/final/jsr052/
- XPath
  www.w3.org/TR/xpath

Parts of this API might be used in the final exam API

© Robert Kelly, 2001-2012

3

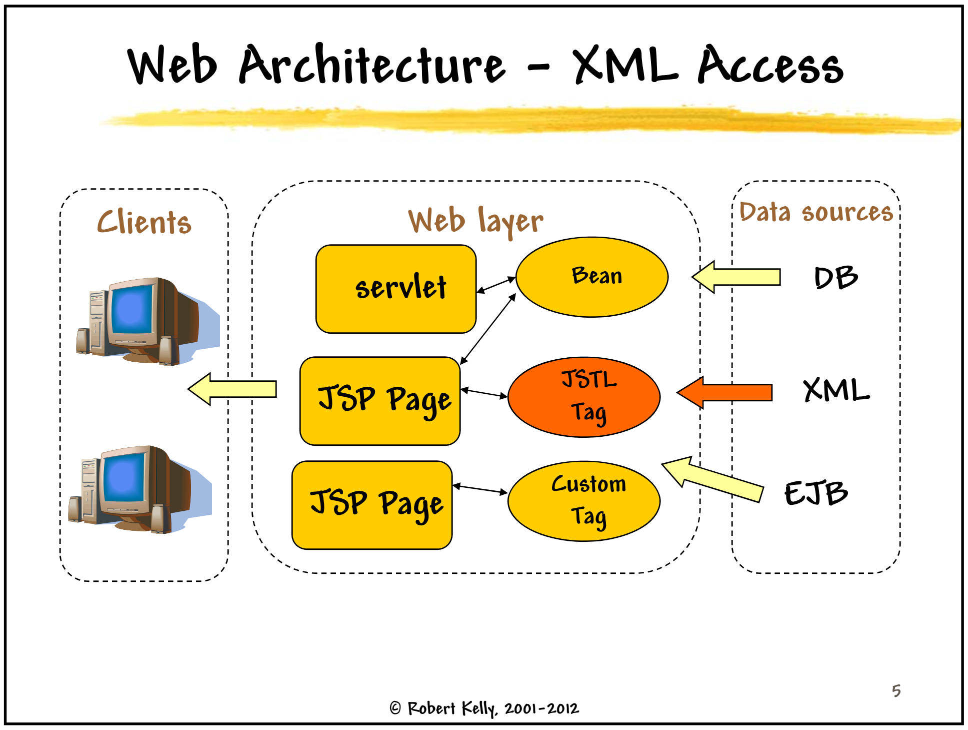


JSTL Versions

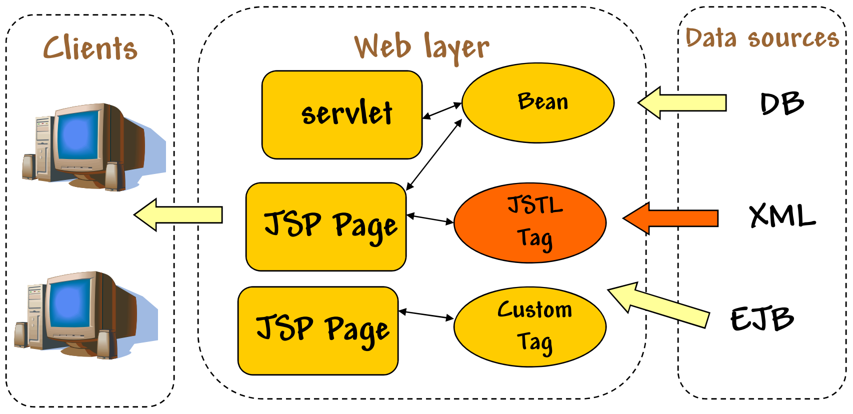
- Versions
  - JSTL 1.0 – requires JSP container that supports the Java Servlet 2.3 and JSP 1.2 specifications
  - JSTL 1.1 – requires a JSP container that supports Java Servlet 2.4 and JSP 2.0
  - JSTL 1.2 – Current release
- Download of JSTL 1.1 available from Apache, but it should already be included in your IDE
  jakarta.apache.org/taglibs/doc/standard-doc/intro.html

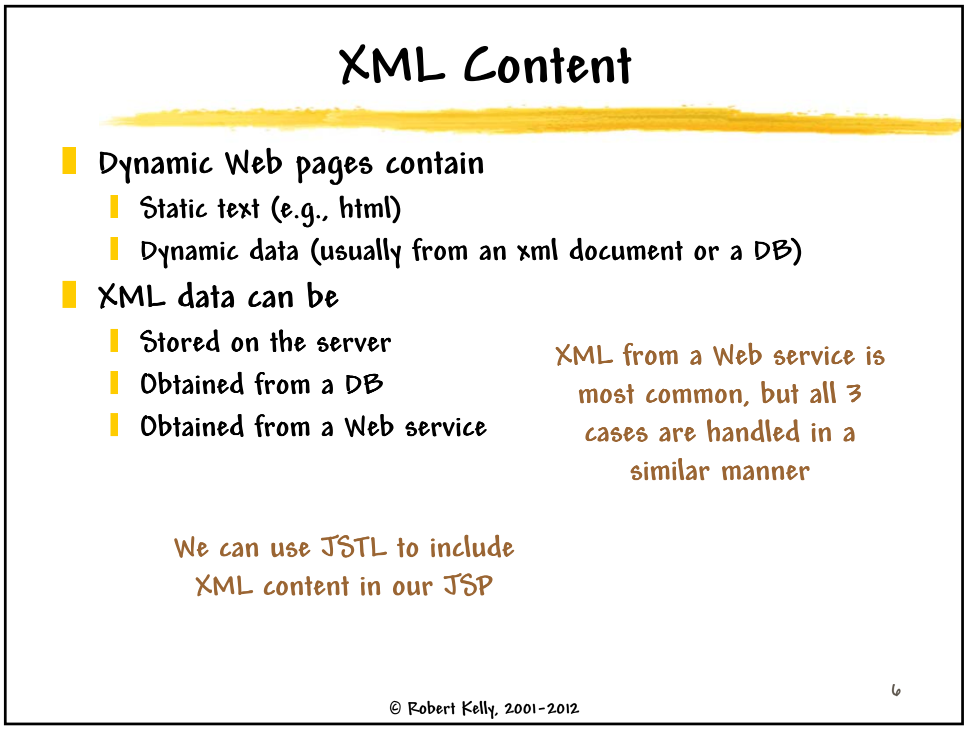
© Robert Kelly, 2001-2012

4



Web Architecture – XML Access



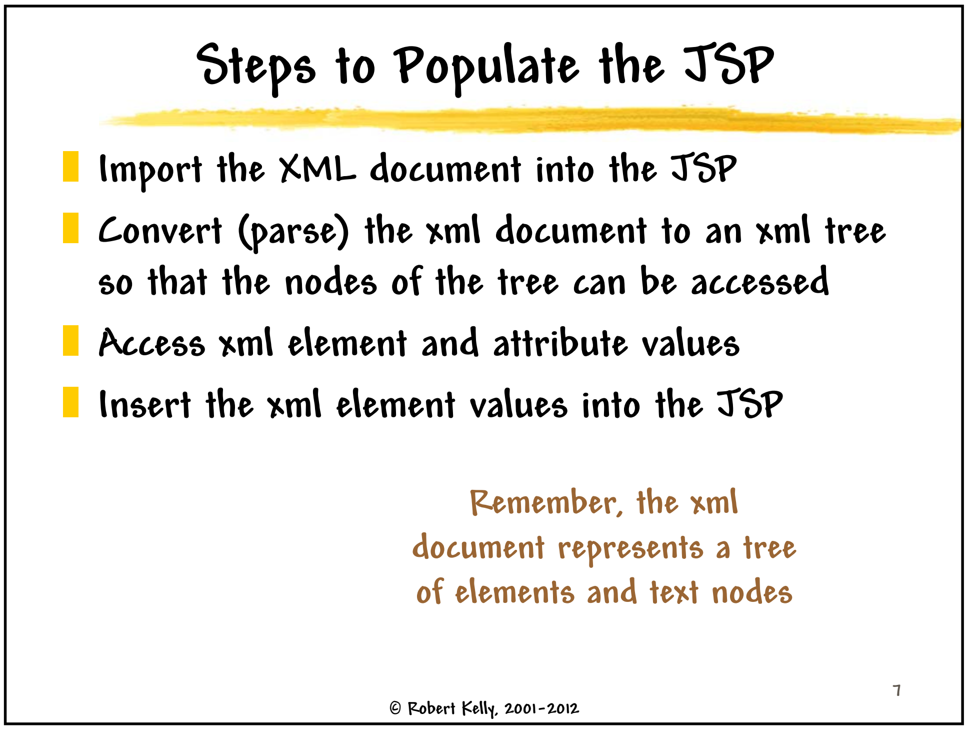


XML Content

- Dynamic Web pages contain
  - Static text (e.g., html)
  - Dynamic data (usually from an xml document or a DB)
- XML data can be
  - Stored on the server
  - Obtained from a DB
  - Obtained from a Web service

XML from a Web service is most common, but all 3 cases are handled in a similar manner

We can use JSTL to include XML content in our JSP

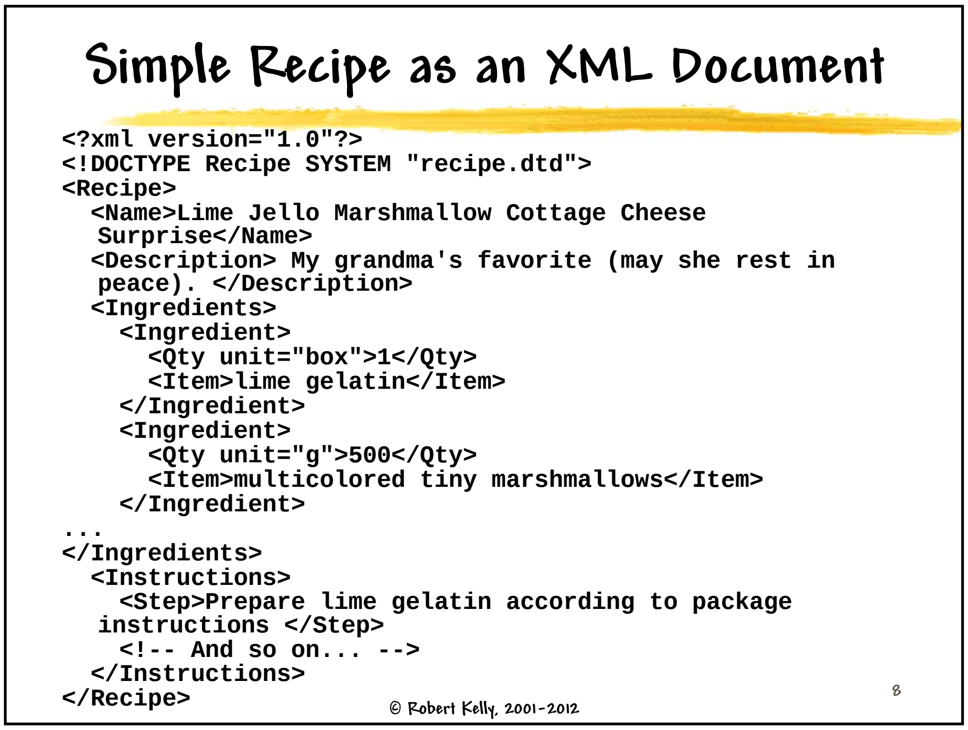


Steps to Populate the JSP

- Import the XML document into the JSP
- Convert (parse) the xml document to an xml tree so that the nodes of the tree can be accessed
- Access xml element and attribute values
- Insert the xml element values into the JSP

Remember, the xml document represents a tree of elements and text nodes

7

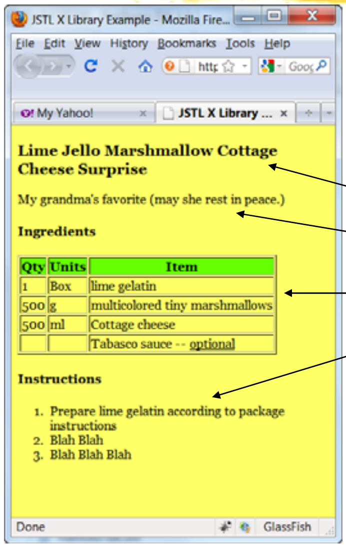


Simple Recipe as an XML Document

```
<?xml version="1.0"?>
<!DOCTYPE Recipe SYSTEM "recipe.dtd">
<Recipe>
  <Name>Lime Jello Marshmallow Cottage Cheese
   Surprise</Name>
  <Description> My grandma's favorite (may she rest in
   peace). </Description>
  <Ingredients>
    <Ingredient>
      <Qty unit="box">1</Qty>
      <Item>lime gelatin</Item>
    </Ingredient>
    <Ingredient>
      <Qty unit="g">500</Qty>
      <Item>multicolored tiny marshmallows</Item>
    </Ingredient>
...
</Ingredients>
  <Instructions>
    <Step>Prepare lime gelatin according to package
   instructions </Step>
    <!-- And so on... -->
  </Instructions>
</Recipe>
```

8

Populating the JSP


JSTL X Library Example - Mozilla Fire...
File Edit View History Bookmarks Tools Help
My Yahoo!
JSTL X Library ...
Done
GlassFish

Lime Jello Marshmallow Cottage Cheese Surprise

My grandma's favorite (may she rest in peace.)

Ingredients

Qty	Units	Item
1	Box	lime gelatin
500	g	multicolored tiny marshmallows
500	ml	Cottage cheese
		Tabasco sauce -- optional

Instructions

1. Prepare lime gelatin according to package instructions
2. Blah Blah
3. Blah Blah Blah

We would like to populate this jsp from the xml document

© Robert Kelly, 2001-2012

9

Import the XML Document

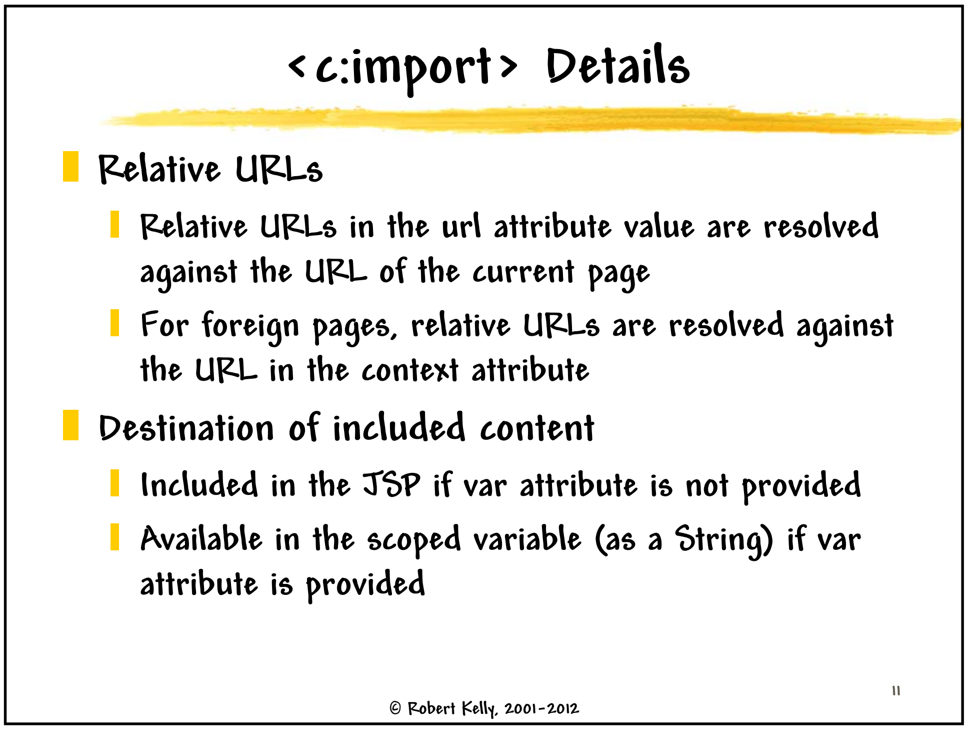
```
<c:import url="recipe-simple.xml" var="x" />
```

- To include a resource from a different Web application (or the same) at run-time, use the import directive
- Attributes
  - url – URL of resource to import
  - var – name of the scoped variable
  - context – context (when accessing a relative resource in a foreign context)
  - charEncoding – character encoding of the import resource
  - scope – standard 4 scopes (default is page context)

Where is "x" stored?

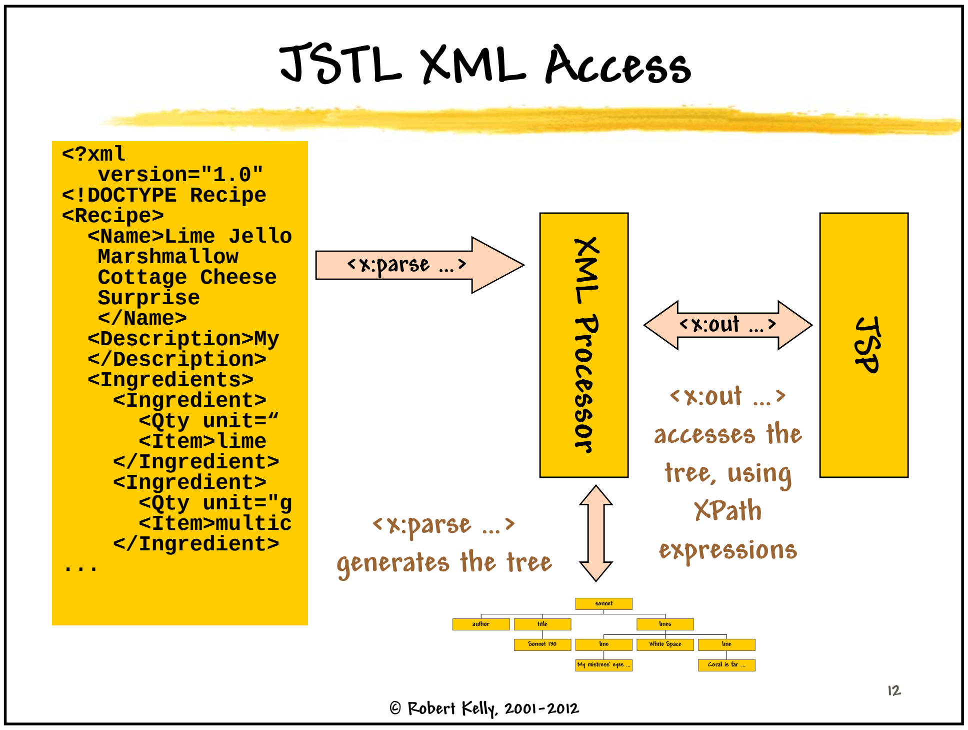
© Robert Kelly, 2001-2012

10



<c:import> Details

- Relative URLs
  - Relative URLs in the url attribute value are resolved against the URL of the current page
  - For foreign pages, relative URLs are resolved against the URL in the context attribute
- Destination of included content
  - Included in the JSP if var attribute is not provided
  - Available in the scoped variable (as a String) if var attribute is provided

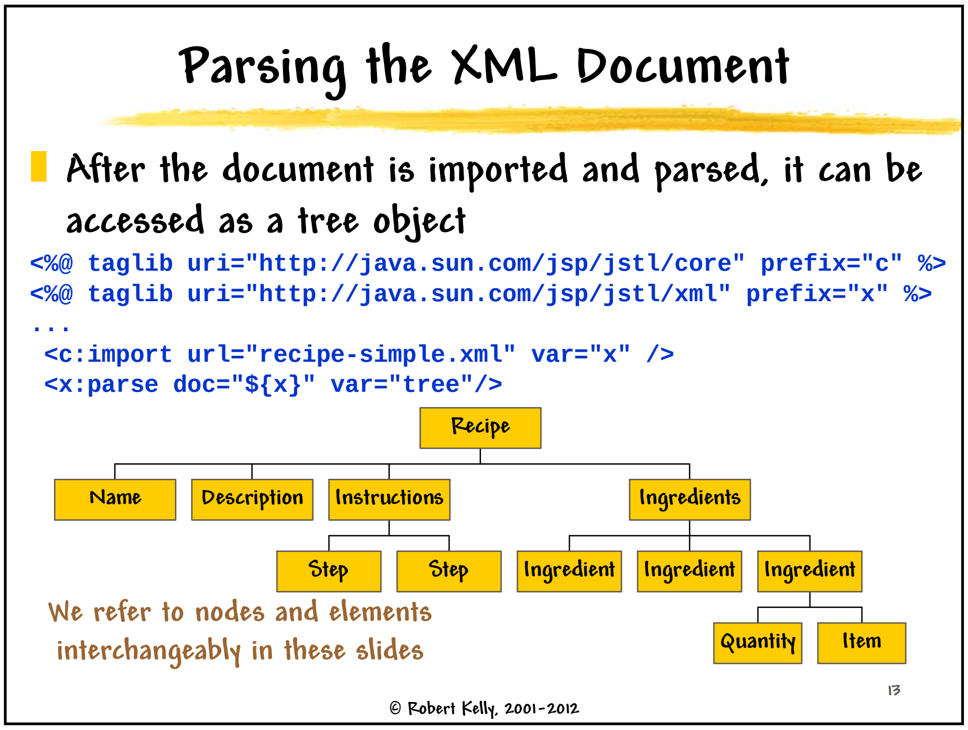


JSTL XML Access

```
<?xml
   version="1.0"
<!DOCTYPE Recipe
<Recipe>
  <Name>Lime Jello
   Marshmallow
   Cottage Cheese
   Surprise
   </Name>
  <Description>My
  </Description>
  <Ingredients>
    <Ingredient>
      <Qty unit="
      <Item>lime
    </Ingredient>
    <Ingredient>
      <Qty unit="g
      <Item>multic
    </Ingredient>
...
```

<x:out ...> accesses the tree, using XPath expressions

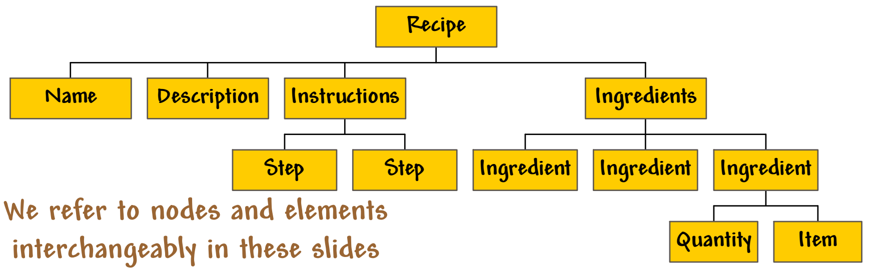
<x:parse ...> generates the tree



Parsing the XML Document

- After the document is imported and parsed, it can be accessed as a tree object

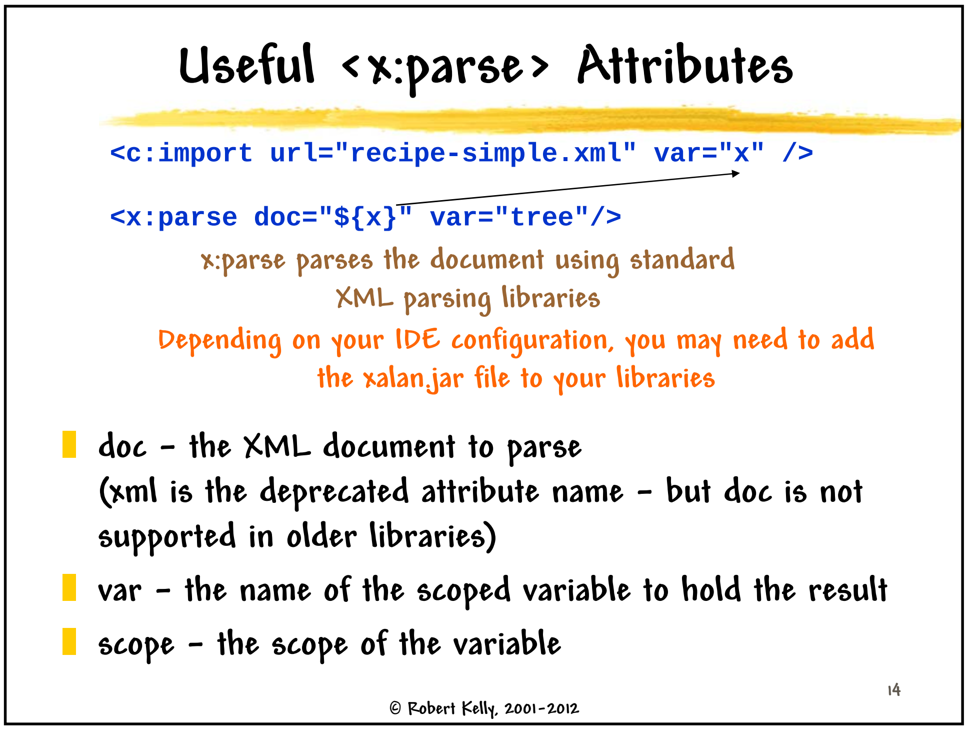
```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x" %>
...
 <c:import url="recipe-simple.xml" var="x" />
 <x:parse doc="${x}" var="tree"/>
```



We refer to nodes and elements interchangeably in these slides

13

© Robert Kelly, 2001-2012



Useful <x:parse> Attributes

```
<c:import url="recipe-simple.xml" var="x" />

<x:parse doc="${x}" var="tree"/>
```

x:parse parses the document using standard XML parsing libraries

Depending on your IDE configuration, you may need to add the xalan.jar file to your libraries

- doc – the XML document to parse
  (xml is the deprecated attribute name – but doc is not supported in older libraries)
- var – the name of the scoped variable to hold the result
- scope – the scope of the variable

14

© Robert Kelly, 2001-2012

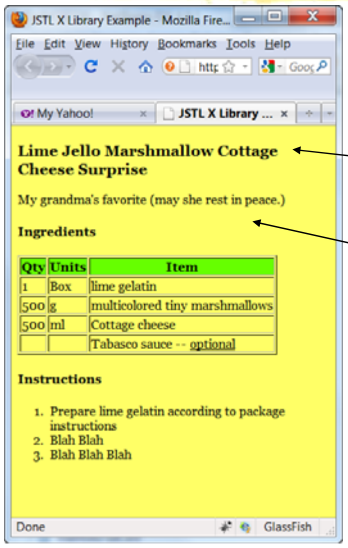
How Do We Get the Data?

- There are 2 major ways to access the data in an XML document and insert into a JSP
  - Custom tag – write a custom tag that uses a Java API to parse and access the XML document
  - JSTL x library

Your custom tag could use one of many XML access APIs

Most accesses can be handled by the JSTL x library

Access the XML Using JSTL


JSTL X Library Example - Mozilla Fire...
File Edit View History Bookmarks Tools Help
My Yahoo!
JSTL X Library ...
Done
GlassFish

Lime Jello Marshmallow Cottage Cheese Surprise

My grandma's favorite (may she rest in peace.)

Ingredients

Qty	Units	Item
1	Box	lime gelatin
500	g	multicolored tiny marshmallows
500	ml	Cottage cheese
		Tabasco sauce -- optional

Instructions

1. Prepare lime gelatin according to package instructions
2. Blah Blah
3. Blah Blah Blah

These are XPath expressions

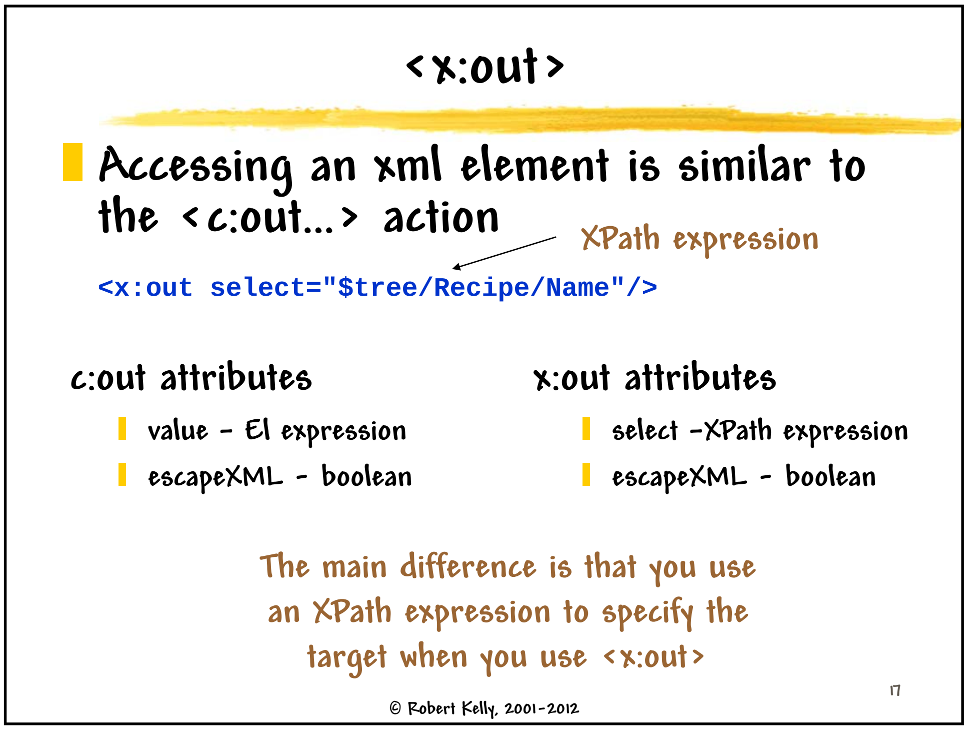
JSP

```
<h3>
<x:out select="$tree/Recipe/Name"/>
</h3>

<p>
<x:out select="$tree//Description"/>
</p>
```

XML Document

```
<Recipe>
 <Name>Lime Jello Marshmallow
Cottage Cheese Surprise </Name>
 <Description>My grandma's favorite
(may she rest in peace.)
</Description>
...
```



<x:out>

- Accessing an xml element is similar to the <c:out…> action

XPath expression

```
<x:out select="$tree/Recipe/Name"/>
```

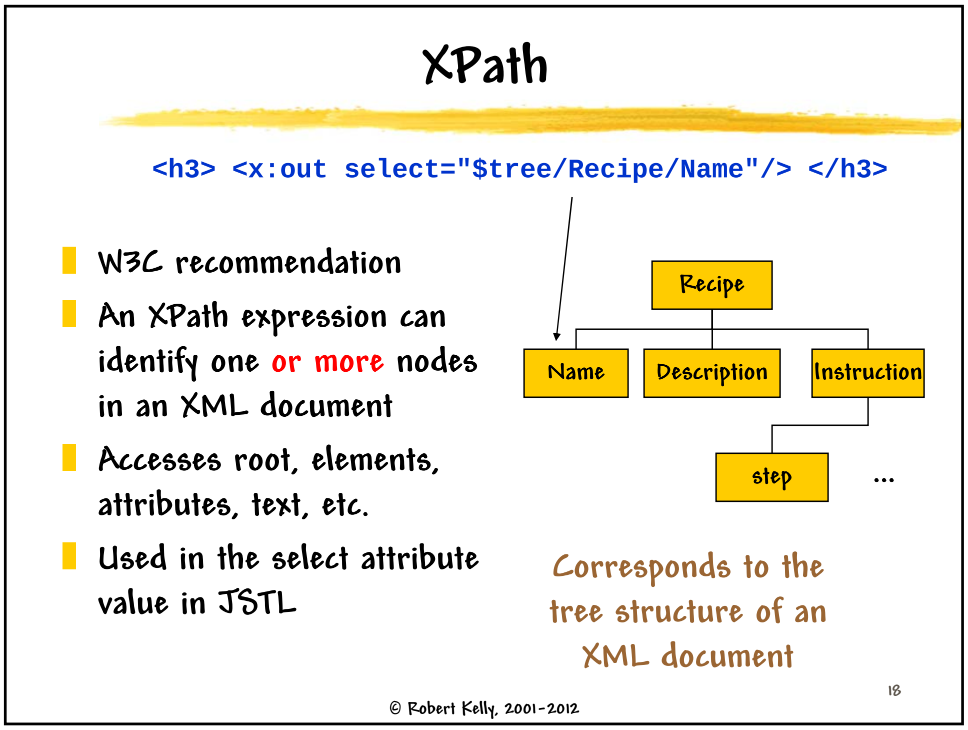
c:out attributes
- value – El expression
- escapeXML - boolean

x:out attributes
- select –XPath expression
- escapeXML - boolean

The main difference is that you use an XPath expression to specify the target when you use <x:out>

© Robert Kelly, 2001-2012

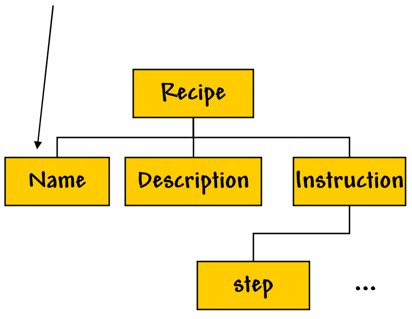
17



XPath

```
<h3> <x:out select="$tree/Recipe/Name"/> </h3>
```

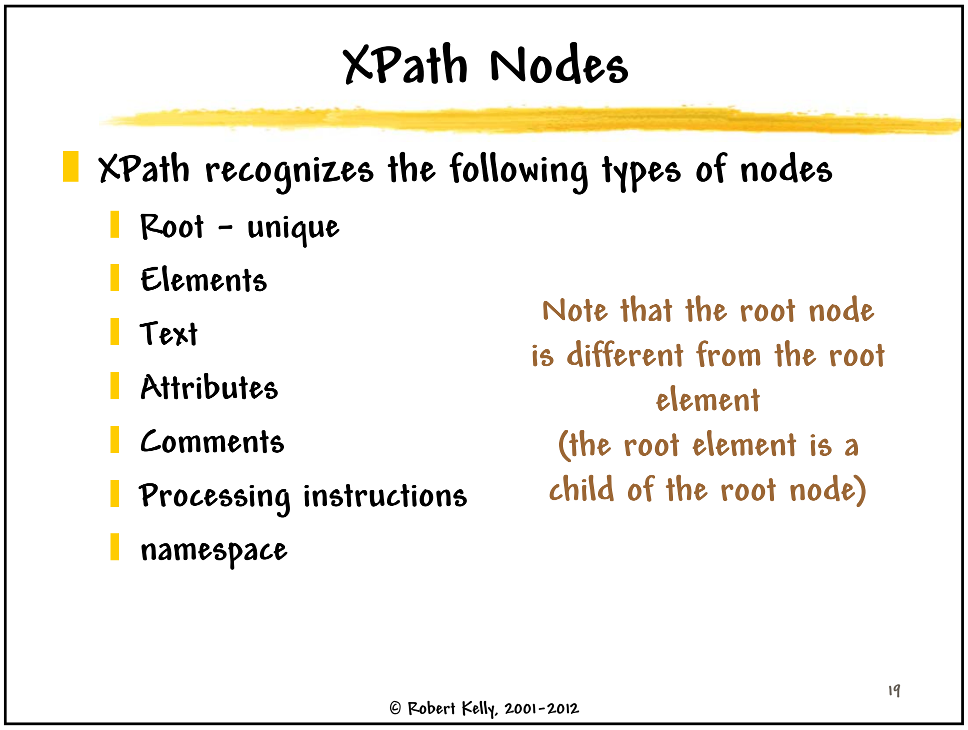
- W3C recommendation
- An XPath expression can identify one or more nodes in an XML document
- Accesses root, elements, attributes, text, etc.
- Used in the select attribute value in JSTL



Corresponds to the tree structure of an XML document

© Robert Kelly, 2001-2012

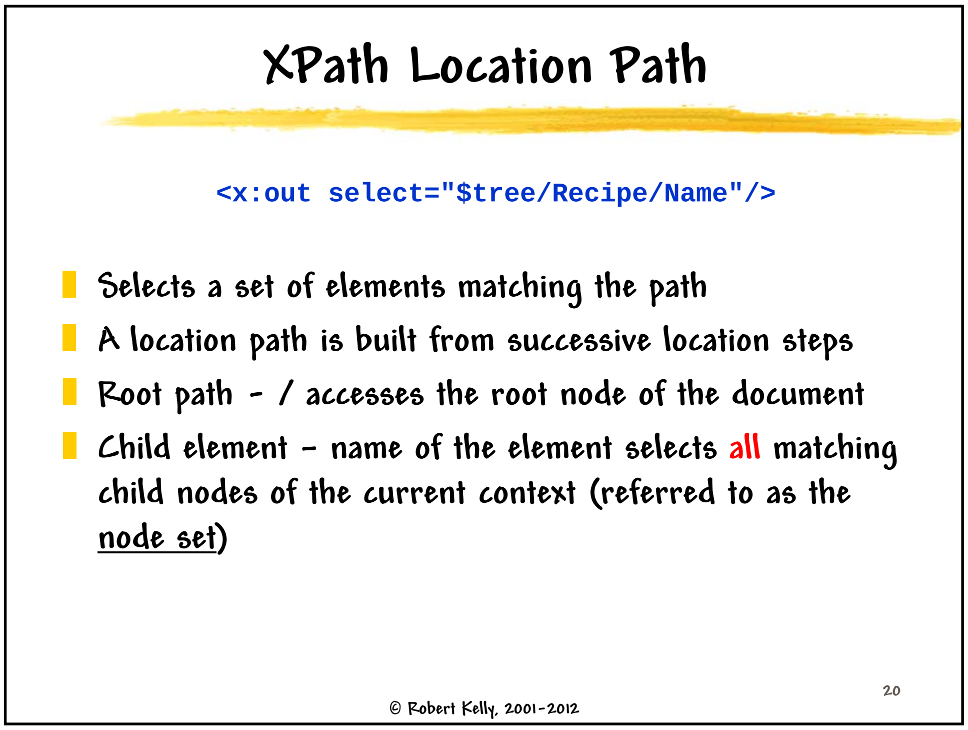
18



XPath Nodes

- XPath recognizes the following types of nodes
  - Root – unique
  - Elements
  - Text
  - Attributes
  - Comments
  - Processing instructions
  - namespace

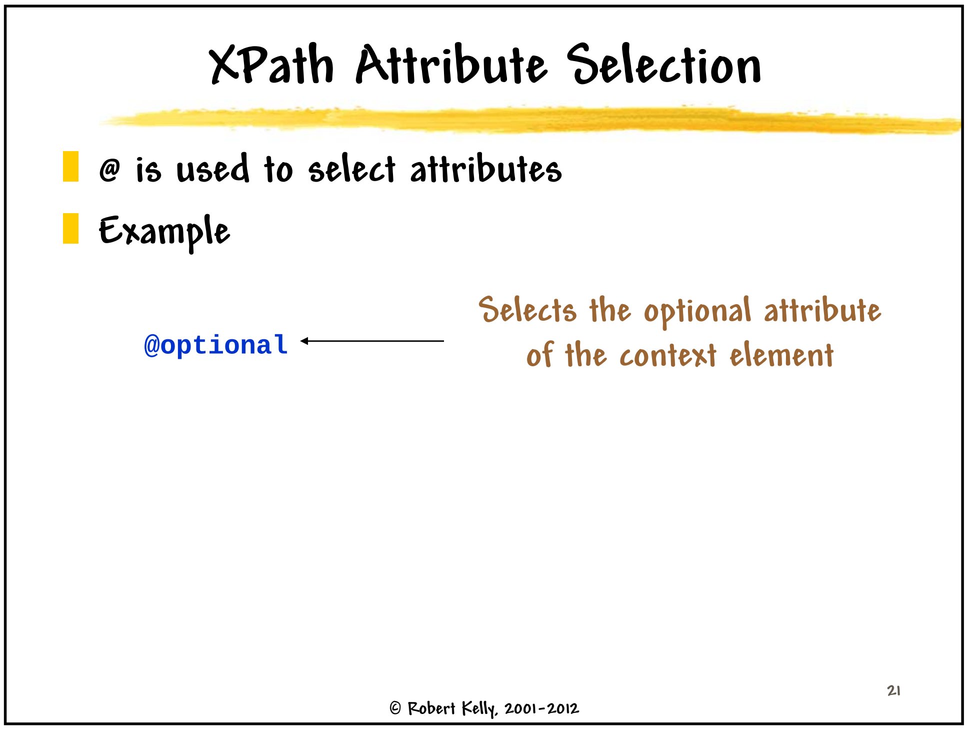
Note that the root node is different from the root element (the root element is a child of the root node)



XPath Location Path

```
<x:out select="$tree/Recipe/Name"/>
```

- Selects a set of elements matching the path
- A location path is built from successive location steps
- Root path – / accesses the root node of the document
- Child element – name of the element selects all matching child nodes of the current context (referred to as the node set)

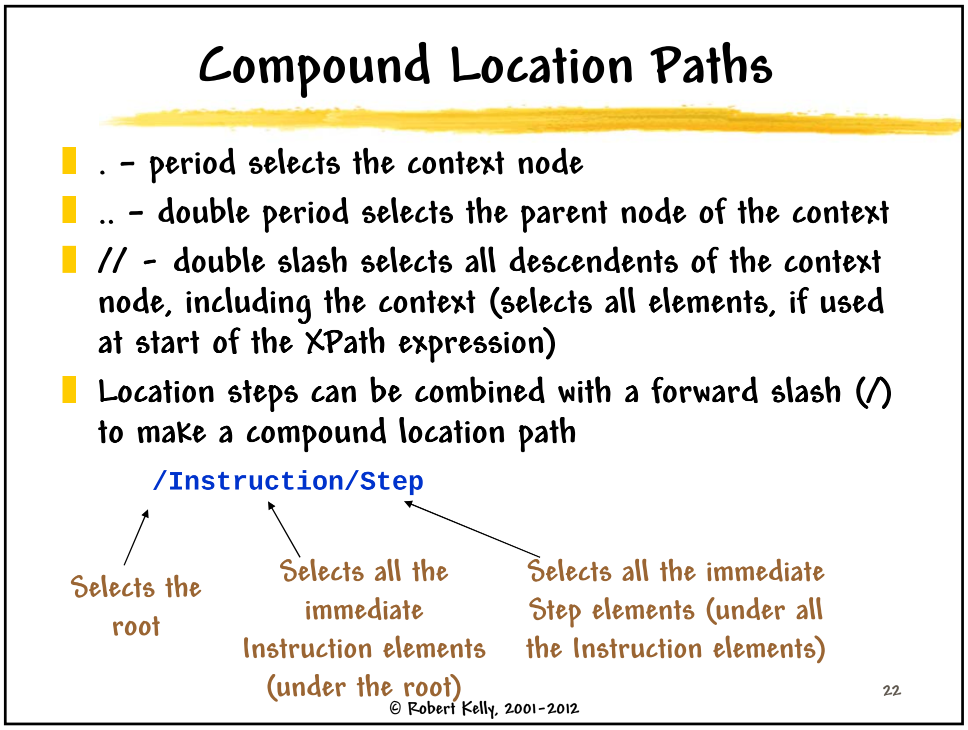


XPath Attribute Selection

- @ is used to select attributes
- Example

```
@optional
```

Selects the optional attribute of the context element



Compound Location Paths

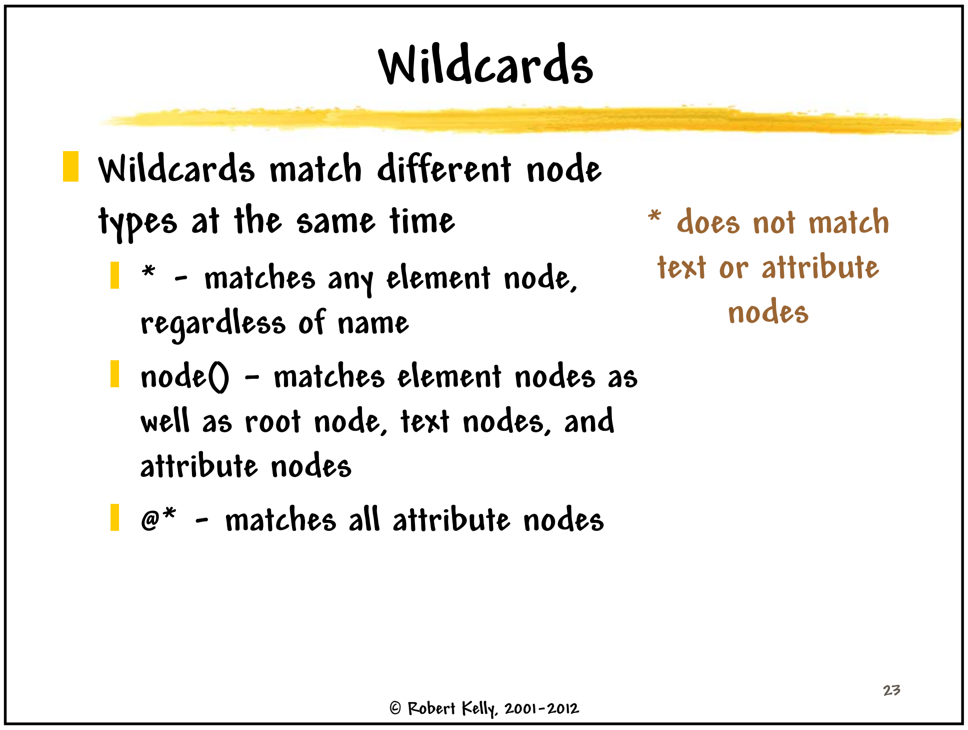
- . – period selects the context node
- .. – double period selects the parent node of the context
- // - double slash selects all descendents of the context node, including the context (selects all elements, if used at start of the XPath expression)
- Location steps can be combined with a forward slash (/) to make a compound location path

```
/Instruction/Step
```

Selects the root

Selects all the immediate Instruction elements (under the root)

Selects all the immediate Step elements (under all the Instruction elements)



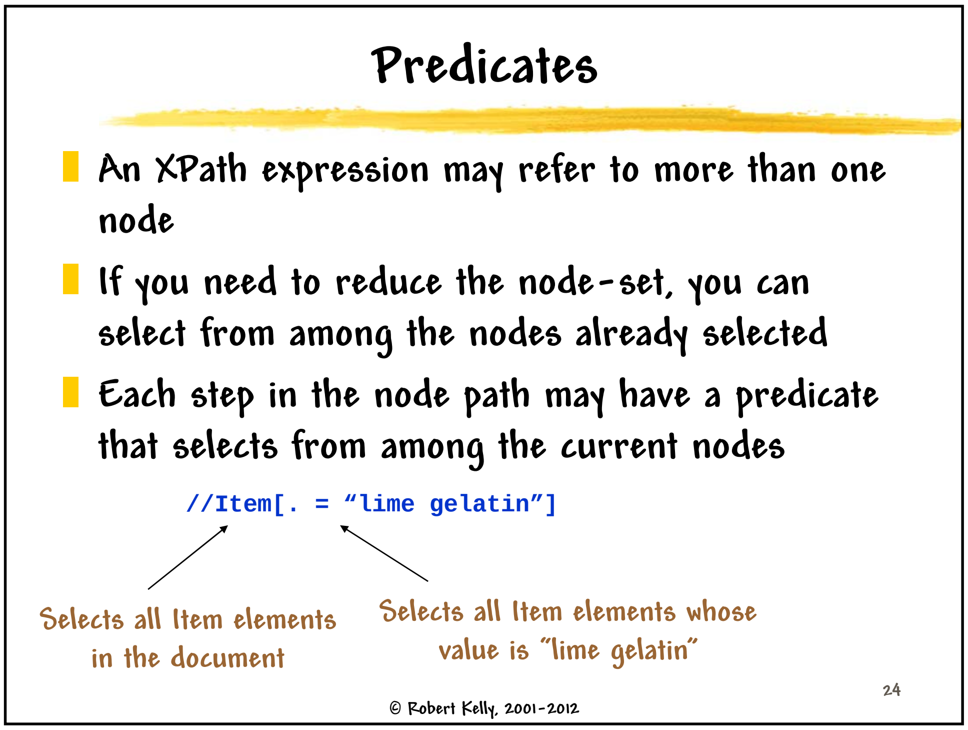
Wildcards

- Wildcards match different node types at the same time
  - * - matches any element node, regardless of name
  - node() - matches element nodes as well as root node, text nodes, and attribute nodes
  - @* - matches all attribute nodes

* does not match text or attribute nodes

© Robert Kelly, 2001-2012

23



Predicates

- An XPath expression may refer to more than one node
- If you need to reduce the node-set, you can select from among the nodes already selected
- Each step in the node path may have a predicate that selects from among the current nodes

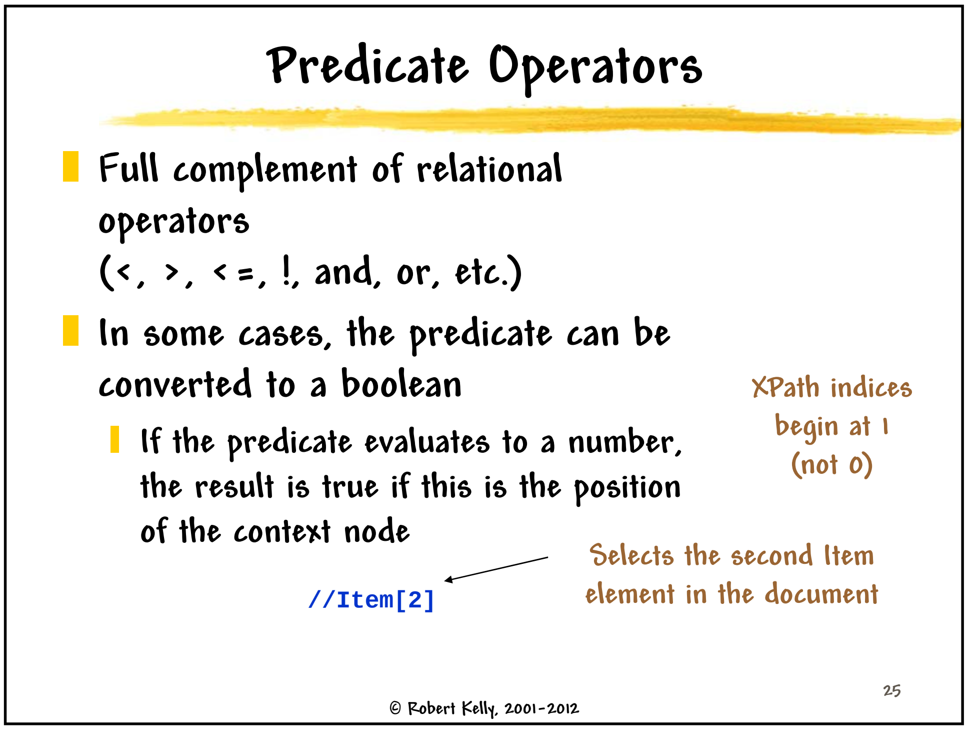
```
//Item[. = “lime gelatin”]
```

Selects all Item elements in the document

Selects all Item elements whose value is “lime gelatin”

© Robert Kelly, 2001-2012

24



Predicate Operators

- Full complement of relational operators
  (<, >, <=, !, and, or, etc.)
- In some cases, the predicate can be converted to a boolean
  - If the predicate evaluates to a number, the result is true if this is the position of the context node

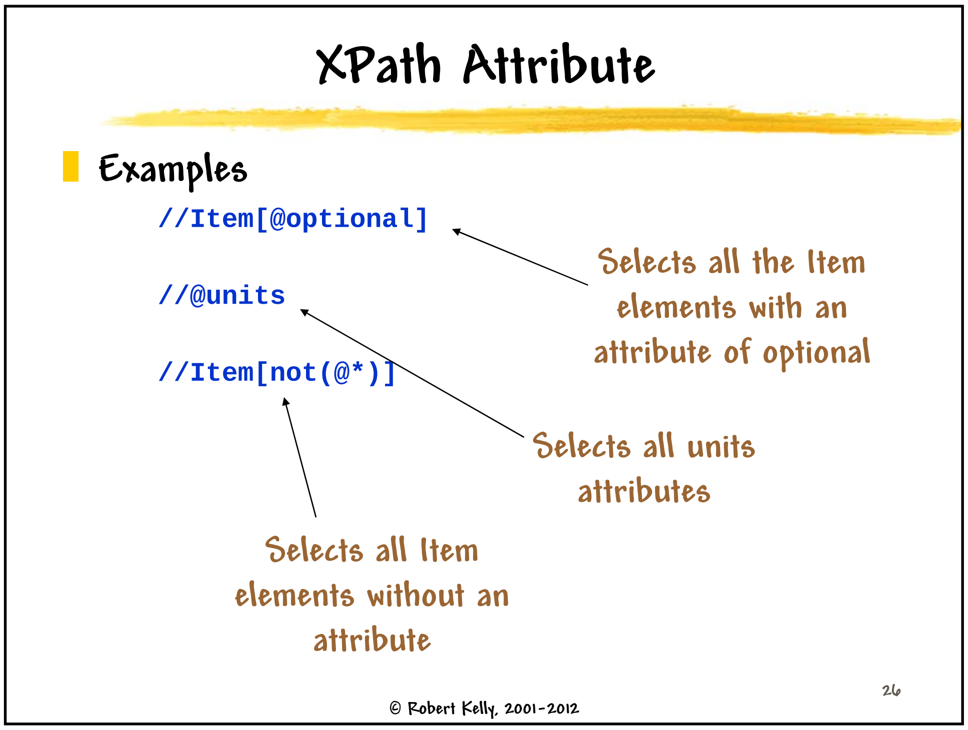
XPath indices begin at 1 (not 0)

```
//Item[2]
```

Selects the second Item element in the document

© Robert Kelly, 2001-2012

25



XPath Attribute

- Examples

```
//Item[@optional]

//@units

//Item[not(@*)]
```

Selects all the Item elements with an attribute of optional

Selects all units attributes

Selects all Item elements without an attribute

© Robert Kelly, 2001-2012

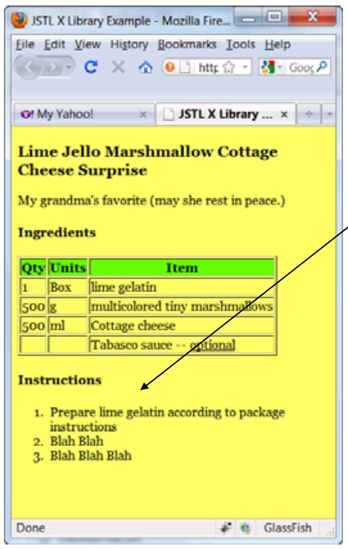
26

Other XPath Functions

- last() - last element in the set
- normalize-space() - removes leading and trailing spaces
- count() - counts the number of elements
- string-length - returns the number of characters in the string

27

How Do We Loop Over Nodes in a JSP?


JSTL X Library Example - Mozilla Fire...
File Edit View History Bookmarks Tools Help
My Yahoo!
JSTL X Library ...
Done
GlassFish

Lime Jello Marshmallow Cottage Cheese Surprise

My grandma's favorite (may she rest in peace.)

Ingredients

Qty	Units	Item
1	Box	lime gelatin
500	g	multicolored tiny marshmallows
500	ml	Cottage cheese
		Tabasco sauce -- optional

Instructions

1. Prepare lime gelatin according to package instructions
2. Blah Blah
3. Blah Blah Blah

<x:forEach allows us to iterate over a collection of nodes

JSP

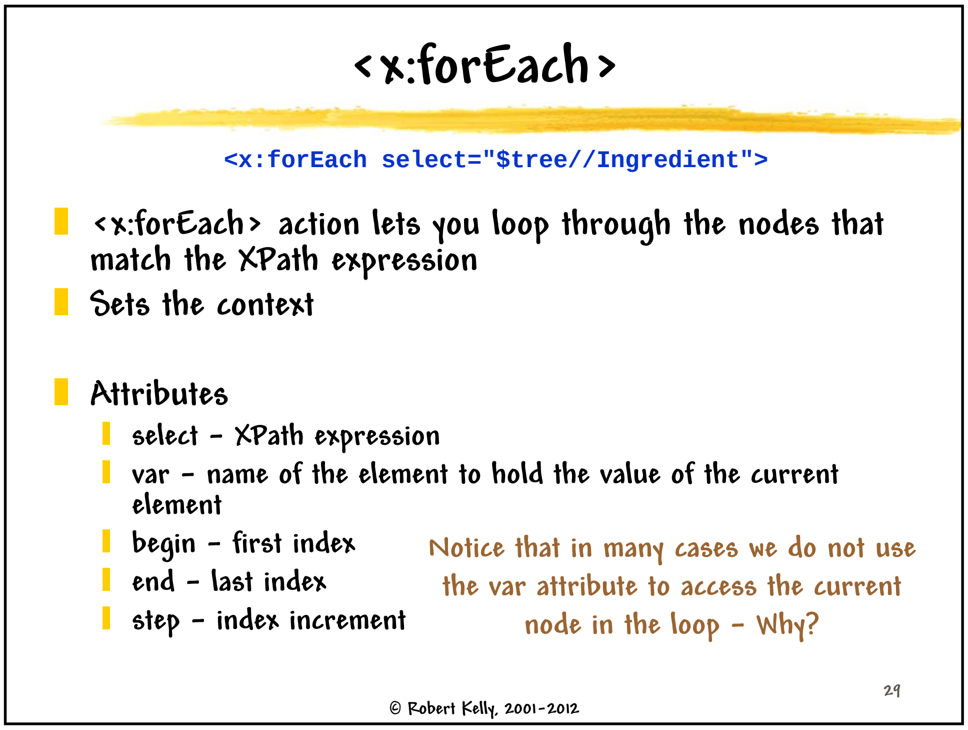
```
<ol>
<x:forEach
select="$tree/Recipe/Instructions/Step">
 <li><x:out select="."/></li>
</x:forEach>
 </ol>
```

Notice the difference compared with c:forEach

XML

```
<Instructions>
 <Step>Prepare lime gelatin according
   to package instructions</Step>
 <Step>Add ingredients</Step>
 <Step>Place in direct sunlight
    for 5 days</Step>
</Instructions>
```

28



<x:forEach>

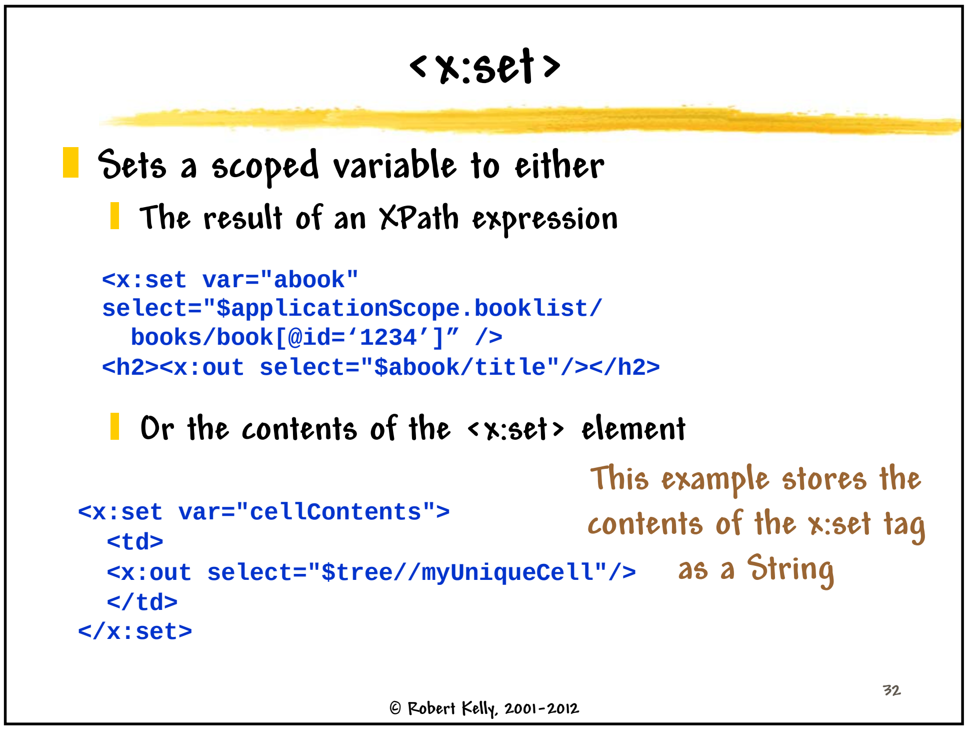
```
<x:forEach select="$tree//Ingredient">
```

- <x:forEach> action lets you loop through the nodes that match the XPath expression
- Sets the context
- Attributes
  - select - XPath expression
  - var - name of the element to hold the value of the current element
  - begin - first index
  - end - last index
  - step - index increment

Notice that in many cases we do not use the var attribute to access the current node in the loop - Why?

© Robert Kelly, 2001-2012

29



<x:set>

- Sets a scoped variable to either
  - The result of an XPath expression

```
<x:set var="abook"
select="$applicationScope.booklist/
  books/book[@id='1234']" />
<h2><x:out select="$abook/title"/></h2>
```

  - Or the contents of the <x:set> element

```
<x:set var="cellContents">
  <td>
  <x:out select="$tree//myUniqueCell"/>
  </td>
</x:set>
```

This example stores the contents of the x:set tag as a String

© Robert Kelly, 2001-2012

32

XML Conditional Logic

```
<x:parse doc="${xmldoc}" var="output“ />
<x:choose>
  <x:when select="$output/portfolio/stock[price > '70']">
  You still have stocks worth over $70.
  </x:when>
  <x:otherwise> You have no stocks worth over $70.
  </x:otherwise>
</x:choose>
```

- Equivalent to the core JSTL library
  - <x:if>
  - <x:choose>
  - <x:when>
  - <x:otherwise>

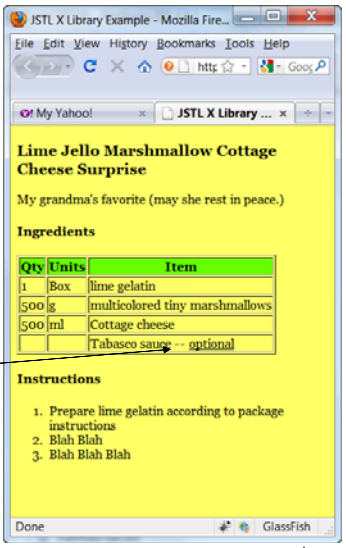
Within an x:choose statement, the x:when statements are evaluated, and the first statement that evaluated to true is processed

33

© Robert Kelly, 2001-2012

Recipe Example

- We produce the recipe page, using JSP and JSTL actions to access elements and attributes in our recipe XML file
- Notice
  - Repeating elements (e.g., ingredients and instructions)
  - Empty table item when Qty is 0
  - The text "optional" is not a text node in the document (the information is in an attribute as an integer)


JSTL X Library Example - Mozilla Fire...
File Edit View History Bookmarks Tools Help
My Yahoo!
JSTL X Library ...
Done
GlassFish

Lime Jello Marshmallow Cottage Cheese Surprise

My grandma's favorite (may she rest in peace.)

Ingredients

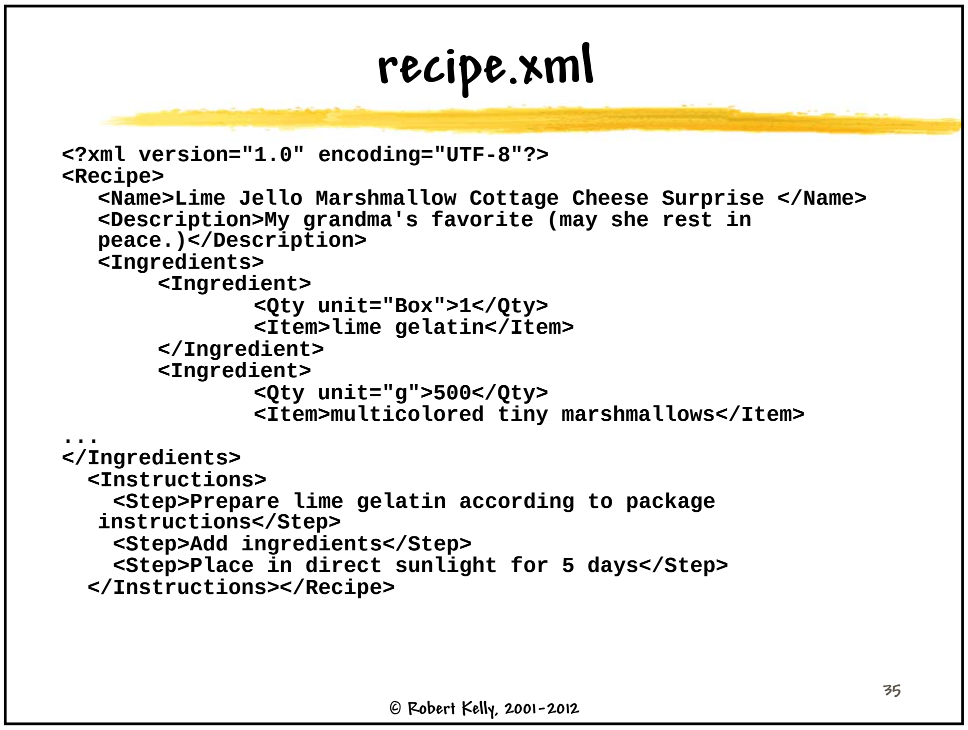
Qty	Units	Item
1	Box	lime gelatin
500	g	multicolored tiny marshmallows
500	ml	Cottage cheese
		Tabasco sauce -- optional

Instructions

1. Prepare lime gelatin according to package instructions
2. Blah Blah
3. Blah Blah Blah

34

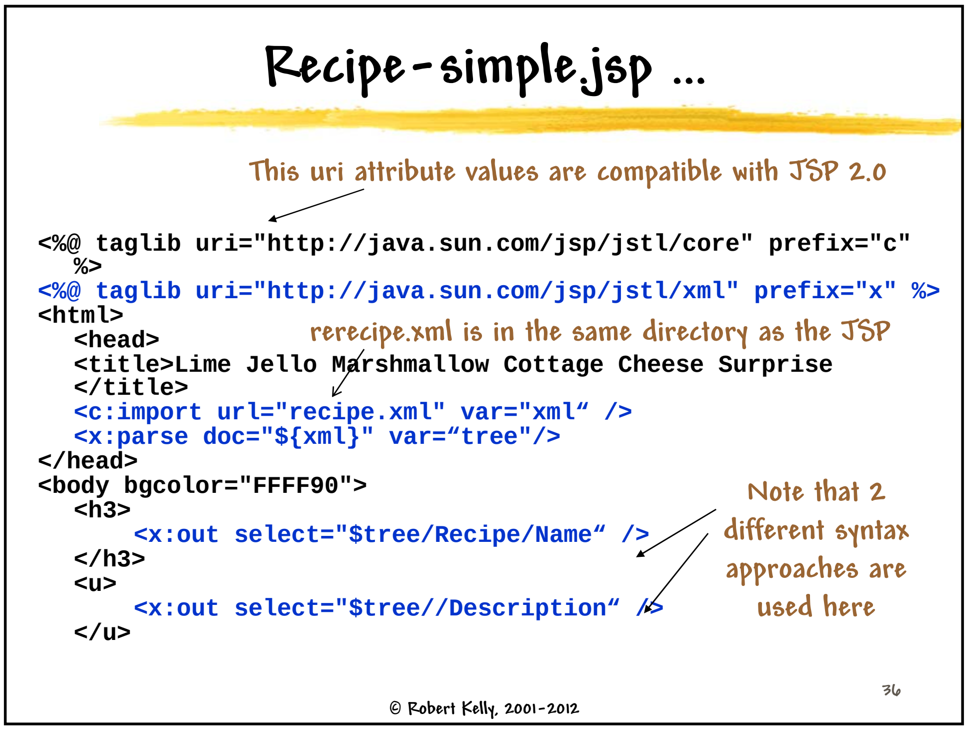
© Robert Kelly, 2001-2012



recipe.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Recipe>
   <Name>Lime Jello Marshmallow Cottage Cheese Surprise </Name>
   <Description>My grandma's favorite (may she rest in
   peace.)</Description>
   <Ingredients>
        <Ingredient>
                <Qty unit="Box">1</Qty>
                <Item>lime gelatin</Item>
        </Ingredient>
        <Ingredient>
                <Qty unit="g">500</Qty>
                <Item>multicolored tiny marshmallows</Item>
...
</Ingredients>
  <Instructions>
    <Step>Prepare lime gelatin according to package
   instructions</Step>
    <Step>Add ingredients</Step>
    <Step>Place in direct sunlight for 5 days</Step>
  </Instructions></Recipe>
```

35



Recipe-simple.jsp ...

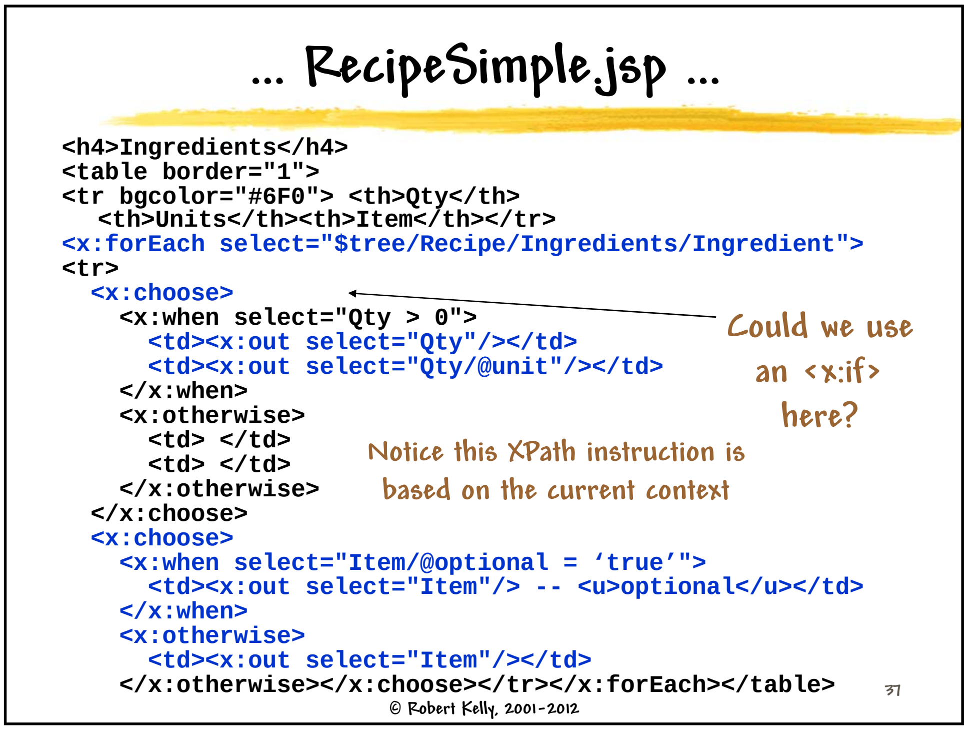
This uri attribute values are compatible with JSP 2.0

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"
   %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x" %>
<html>
   <head>
   <title>Lime Jello Marshmallow Cottage Cheese Surprise
   </title>
   <c:import url="recipe.xml" var="xml“ />
   <x:parse doc="${xml}" var=“tree"/>
</head>
<body bgcolor="FFFF90">
   <h3>
        <x:out select="$tree/Recipe/Name“ />
   </h3>
   <u>
        <x:out select="$tree//Description“ />
   </u>
```

rerecipe.xml is in the same directory as the JSP

Note that 2 different syntax approaches are used here

36



… RecipeSimple.jsp …

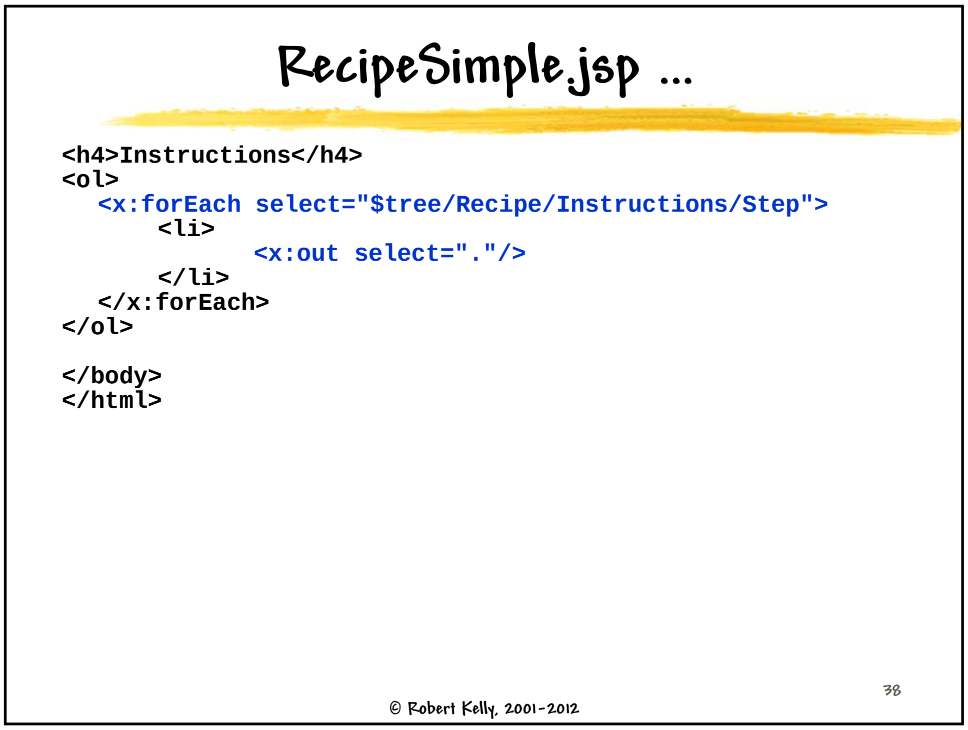
```
<h4>Ingredients</h4>
<table border="1">
<tr bgcolor="#6F0"> <th>Qty</th>
  <th>Units</th><th>Item</th></tr>
<x:forEach select="$tree/Recipe/Ingredients/Ingredient">
<tr>
  <x:choose>
    <x:when select="Qty > 0">
      <td><x:out select="Qty"/></td>
      <td><x:out select="Qty/@unit"/></td>
    </x:when>
    <x:otherwise>
      <td> </td>
      <td> </td>
    </x:otherwise>
  </x:choose>
  <x:choose>
    <x:when select="Item/@optional = ‘true’">
      <td><x:out select="Item"/> -- <u>optional</u></td>
    </x:when>
    <x:otherwise>
      <td><x:out select="Item"/></td>
    </x:otherwise></x:choose></tr></x:forEach></table>
```

Could we use an <x:if> here?

Notice this XPath instruction is based on the current context

37

© Robert Kelly, 2001-2012



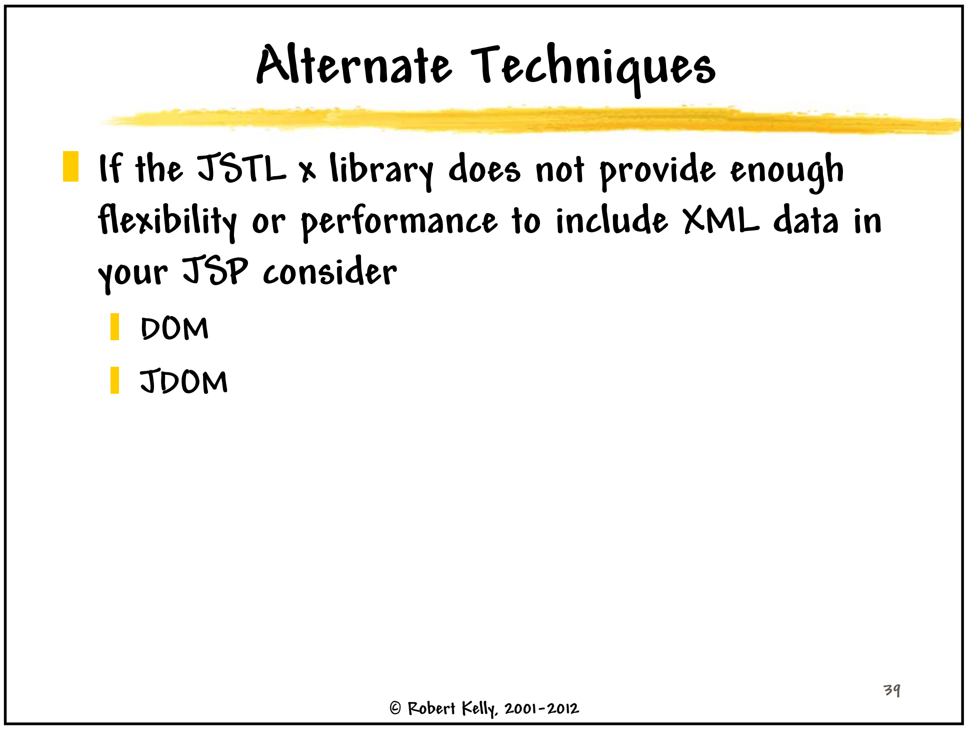
RecipeSimple.jsp …

```
<h4>Instructions</h4>
<ol>
  <x:forEach select="$tree/Recipe/Instructions/Step">
      <li>
            <x:out select="."/>
      </li>
  </x:forEach>
</ol>

</body>
</html>
```

38

© Robert Kelly, 2001-2012

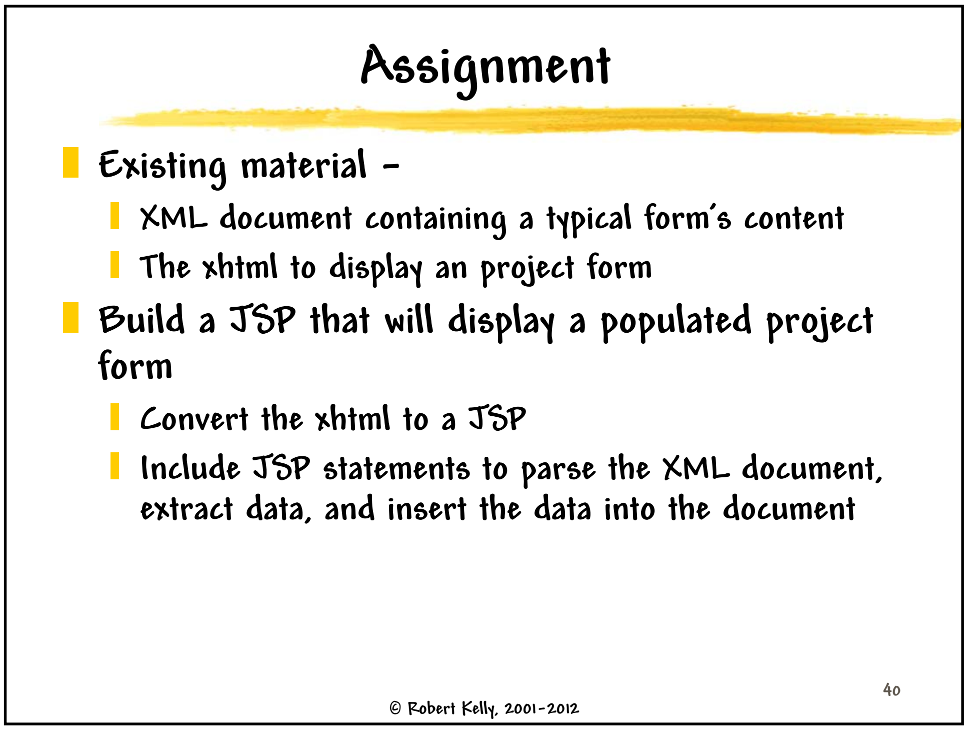


Alternate Techniques

- If the JSTL x library does not provide enough flexibility or performance to include XML data in your JSP consider
  - DOM
  - JDOM

© Robert Kelly, 2001-2012

39



Assignment

- Existing material -
  - XML document containing a typical form's content
  - The xhtml to display an project form
- Build a JSP that will display a populated project form
  - Convert the xhtml to a JSP
  - Include JSP statements to parse the XML document, extract data, and insert the data into the document

© Robert Kelly, 2001-2012

40