

Session 16

XPath

1

Objectives

- Understand XPath well enough to provide a background to jQuery

© Robert Kelly, 2016-2018

2

Reading and References

■ Reading

■ Accessing XML Content

<http://en.wikipedia.org/wiki/Xpath>

https://www.w3schools.com/xml/xpath_intro.asp

■ Reference - Xpath

www.w3.org/TR/xpath

© Robert Kelly, 2016-2018

3

XPath

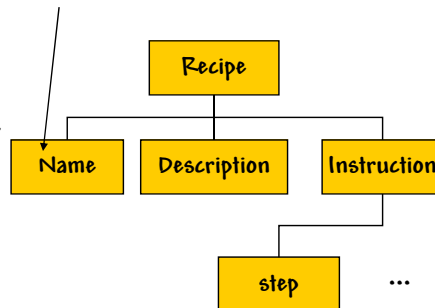
```
<h3> <x:out select="$tree/Recipe/Name"/> </h3>
```

■ 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 X Library



Corresponds to the tree structure of an XML document

© Robert Kelly, 2016-2018

4

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)

© Robert Kelly, 2016-2018

5

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)

© Robert Kelly, 2016-2018

6

XPath Attribute Selection

- @ is used to select attributes
- Example

@optional

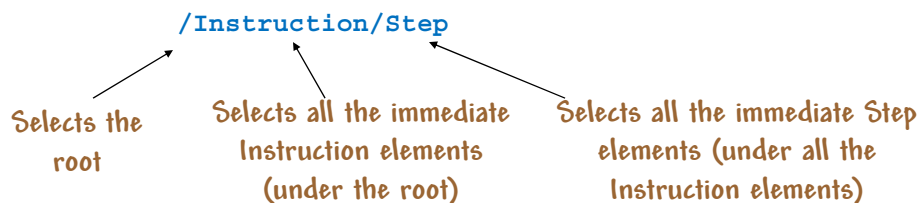
Selects the optional attribute
of the context element

© Robert Kelly, 2016-2018

7

Compound Location Paths

- . - period selects the context node
- .. - double period selects the parent node of the context
- // - double slash selects all descendants 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



© Robert Kelly, 2016-2018

8

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

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"

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, 2016-2018

11

XPath Attribute

■ Examples

`//Item[@optional]`

Selects all the Item
elements with an
attribute of optional

`//@units`

Selects all units
attributes

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

Selects all Item elements without
an attribute

© Robert Kelly, 2016-2018

12

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

© Robert Kelly, 2016-2018

13

Did You Satisfy the Objective?

- Understand XPath well enough to provide a background to jQuery

© Robert Kelly, 2016-2018

14