

XPath - XML Path Language

XML



NIIT
No.1 IT Institute in Asia

<http://yht4ever.blogspot.com>

thanh.phamhong@niithoasen.com
B070066 - NIIT Quang Trung

08/2007

Contents



Introduction



Nodes



Location Paths



Node-set Operators and Functions



Exercises



❖ XML Path Language (XPath)

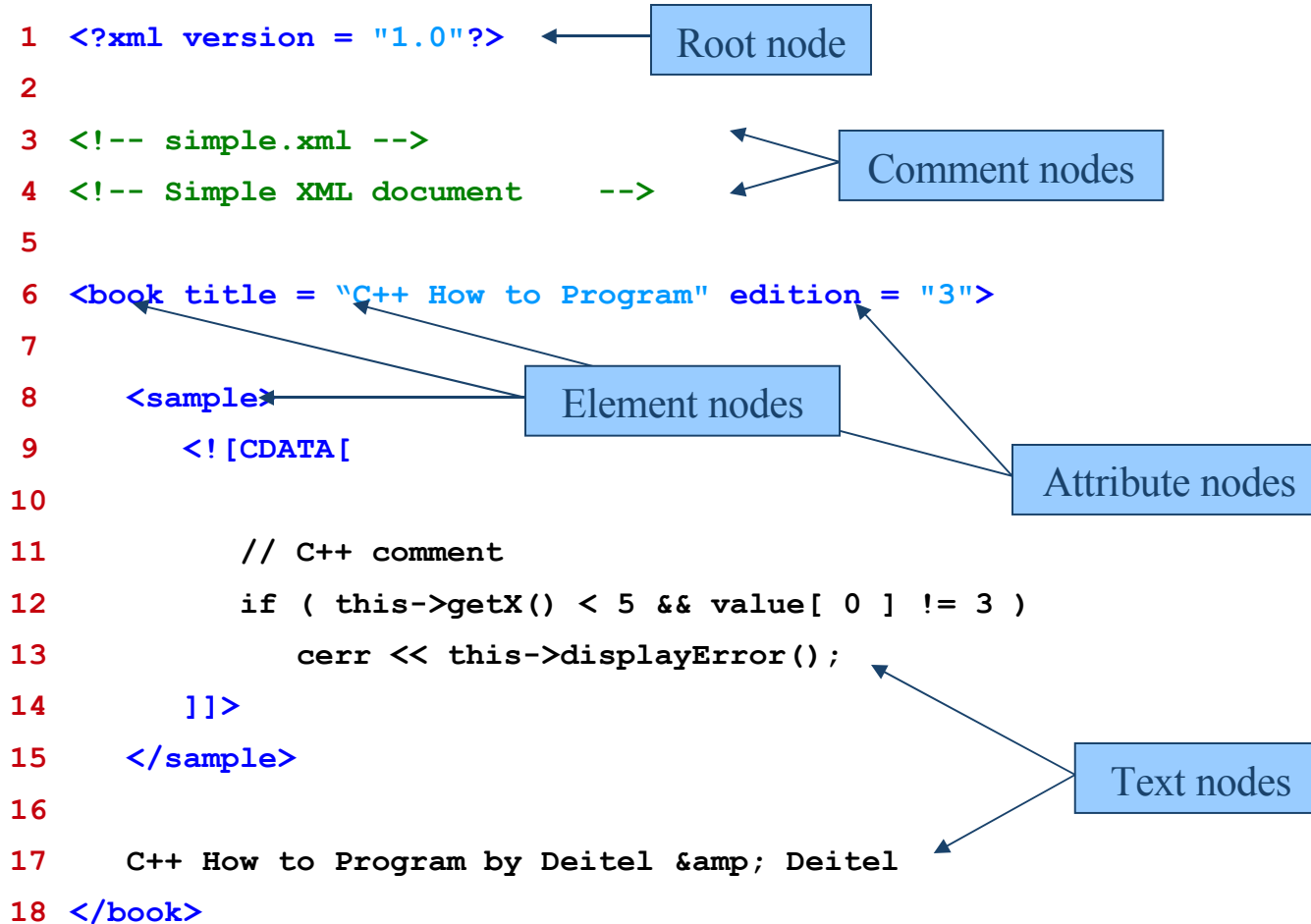
- Syntax for locating information in XML document
 - e.g., attribute values
- String-based language of expressions
 - Not structural language like XML
- Used by other XML technologies
 - XSLT
 - XPointer



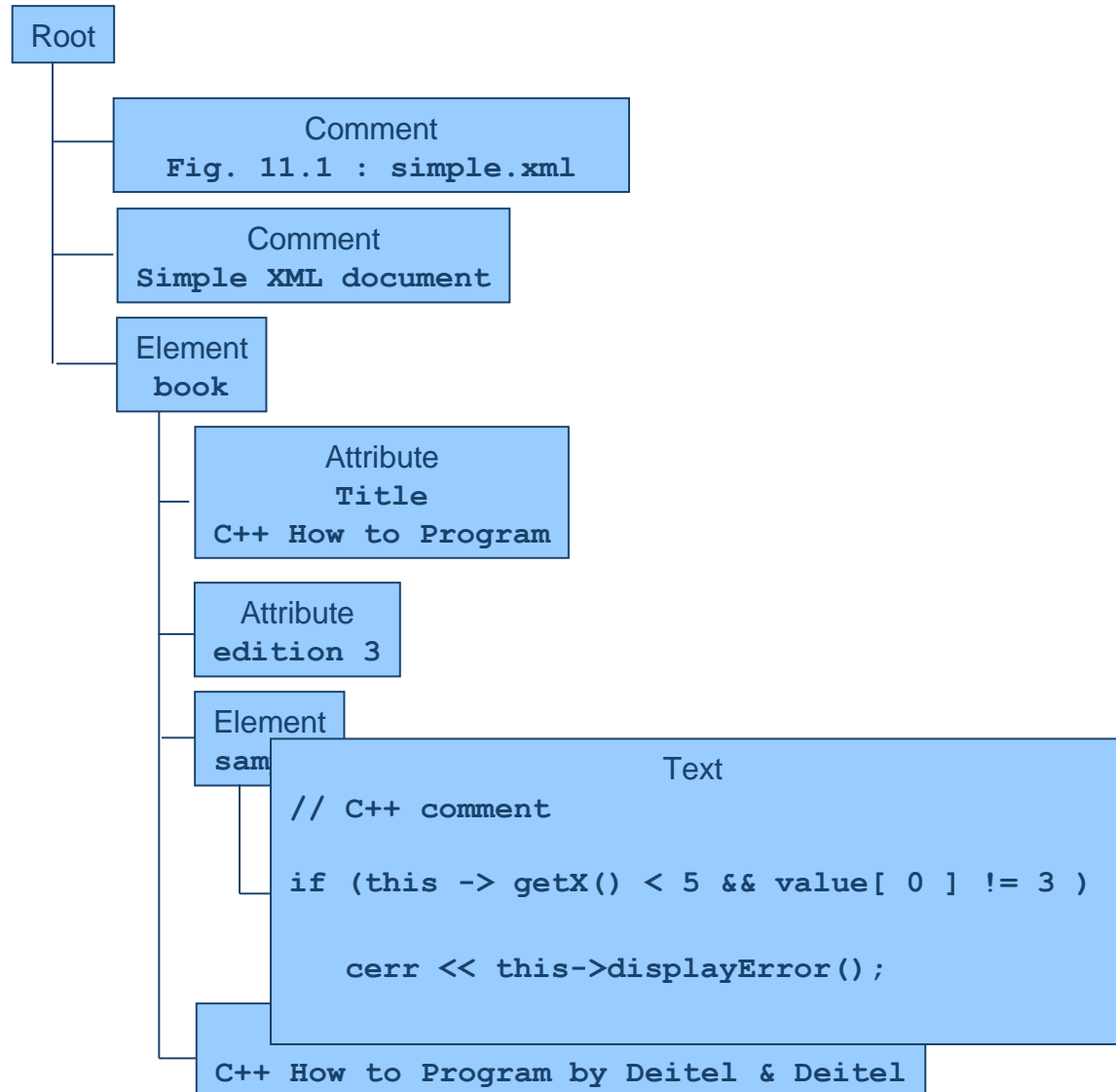
❖ XML document

- Tree structure with nodes
- Each node represents part of XML document
 - Root
 - Element
 - Attribute
 - Text
 - Comment
 - Processing instruction
 - Namespace
- Attributes and namespaces are not children of their parent node
 - They describe their parent node

Simple XML document



XPath tree for simple.xml



XML document with processing-instruction and namespace nodes



```
1  <?xml version = "1.0"?>
2
3  <!-- Fig. 11.3 : simple2.xml
4  <!-- Processing instructions and namespaces
5
6  <html xmlns = "http://www.w3.org/TR/REC-html40">
7
8  <head>
9      <title>Processing Instruction and Namespace Nodes</title>
10 </head>
11
12 <?deitelprocessor example = "fig11_03.xml"?>
13
14 <body>
15
16     <deitel:book deitel:edition = "1"
17         xmlns:deitel = "http://www.deitel.com/xmlhttp1">
18         <deitel:title>XML How to Program</deitel:title>
19     </deitel:book>
20
21 </body>
22
23 </html>
```

Root node

Comment nodes

Namespace nodes

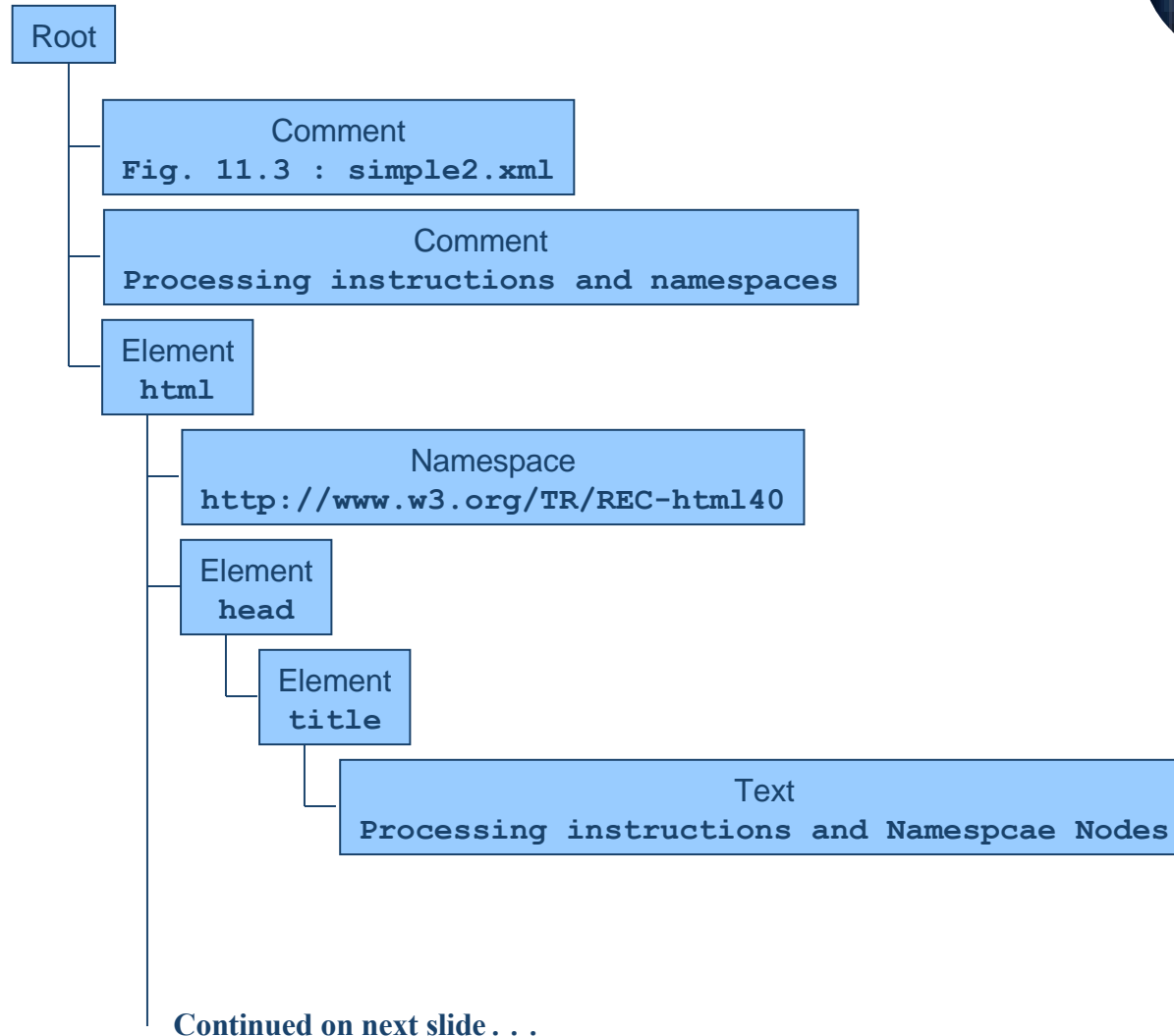
Processing instruction node

Element nodes

Text nodes

Attribute nodes

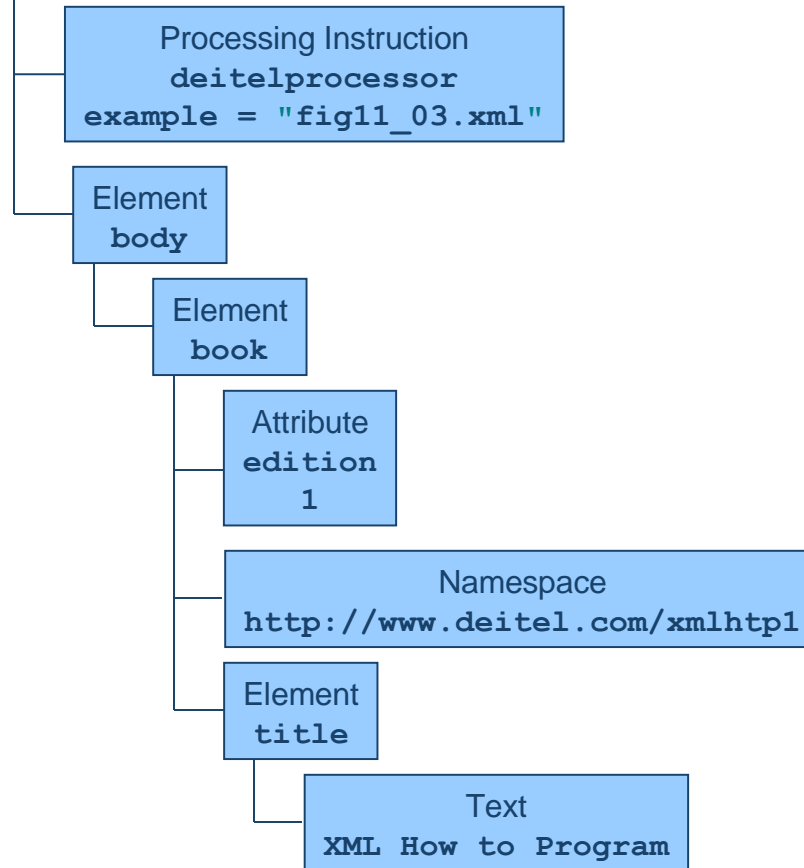
Tree diagram of an XML document with a processing-instruction node



Tree diagram of an XML document with a processing-instruction node



Continued from previous slide



XPath node types



Node Type	string-value	expanded-name	Description
root	Determined by concatenating the string-values of all text-node descendents in document order.	None.	Represents the root of an XML document. This node exists only at the top of the tree and may contain element, comment or processor-instruction children.
element	Determined by concatenating the string-values of all text-node descendents in document order.	The element tag, including the namespace prefix (if applicable).	Represents an XML element and may contain element, text, comment or processor-instruction children.
attribute	The normalized value of the attribute.	The name of the attribute, including the namespace prefix (if applicable).	Represents an attribute of an element.

XPath node types (cont.)



Node Type	string-value	expanded-name	Description
text	The character data contained in the text node.	None.	Represents the character data content of an element.
comment	The content of the comment (not including <code><!--</code> and <code>--></code>).	None.	Represents an XML comment.
processing instruction	The part of the processing instruction that follows the target and any whitespace.	The target of the processing instruction.	Represents an XML processing instruction.
namespace	The URI of the namespace.	The namespace prefix.	Represents an XML namespace.



❖ Location path

- Expression specifying how to navigate XPath tree
- Composed of *location steps*
 - Each location step composed of
 - Axis (Mối liên hệ)
 - Node test
 - Predicate



❖ **XPath searches are made relative to *context node***

❖ **Axis**

- Indicates which nodes are included in search
 - Relative to context node
- Dictates node ordering in set
 - Forward axes select nodes that follow context node
 - Reverse axes select nodes that precede context node

XPath axes



Axis Name	Ordering	Description
self	none	The context node itself.
parent	reverse	The context node's parent, if one exists.
child	forward	The context node's children, if they exist.
ancestor	reverse	The context node's ancestors, if they exist.
ancestor-or-self	reverse	The context node's ancestors and also itself.
descendant	forward	The context node's descendants.
descendant-or-self	forward	The context node's descendants and also itself.
following	forward	The nodes in the XML document following the context node, not including descendants.
following-sibling	forward	The sibling nodes following the context node.
preceding	reverse	The nodes in the XML document preceding the context node, not including ancestors.
preceding-sibling	reverse	The sibling nodes preceding the context node.
attribute	forward	The attribute nodes of the context node.
namespace	forward	The namespace nodes of the context node.



❖ Node tests

- Refine set of nodes selected by axis
 - Rely upon axis' *principle node type*
 - Corresponds to type of node axis can select

Some XPath node tests



Node Test	Description
*	Selects all nodes of the same principal node type.
node ()	Selects all nodes, regardless of their type.
text ()	Selects all text nodes.
comment ()	Selects all comment nodes.
processing-instruction ()	Selects all processing-instruction nodes.
<i>node name</i>	Selects all nodes with the specified <i>node name</i> .

Location Paths Using Axes and Node Tests



❖ Location step

- Axis and node test separated by double colon (: :)
- Optional *predicate* enclosed in square brackets ([])
- Some examples:
 - Select all element-node children of context node
`child::*`
 - Select all text-node children of context node
`child::text()`
 - Select all text-node grandchildren of context node
`child::* / child::text()`

Some location-path abbreviations



Location Path	Description
<code>child::</code>	This location path is used by default if no axis is supplied and may therefore be omitted.
<code>attribute::</code>	The attribute axis may be abbreviated as <code>@</code> .
<code>/descendant-or-self::node()</code>	This location path is abbreviated as two slashes (<code>//</code>).
<code>self::node()</code>	The context node is abbreviated with a period (<code>.</code>).
<code>parent::node()</code>	The context node's parent is abbreviated with two periods (<code>..</code>).



❖ Node-set operators

- Manipulate node sets to form others

❖ Node-set functions

- Perform actions on node-sets returned by location paths

Node-set operators



Node-set Operators	Description
pipe ()	Performs the union of two node-sets.
slash (/)	Separates location steps.
double-slash (//)	Abbreviation for the location path <code>/descendant-or-self::node()</code> /

Some node-set functions



Node-set Functions	Description
last()	Returns the number of nodes in the node-set.
position()	Returns the position number of the current node in the node-set being tested.
count(<i>node-set</i>)	Returns the number of nodes in <i>node-set</i> .
id(<i>string</i>)	Returns the element node whose ID attribute matches the value specified by argument <i>string</i> .
local-name(<i>node-set</i>)	Returns the local part of the expanded-name for the first node in <i>node-set</i> .
namespace-uri(<i>node-set</i>)	Returns the namespace URI of the expanded-name for the first node in <i>node-set</i> .
name(<i>node-set</i>)	Returns the qualified name for the first node in <i>node-set</i> .

Node-set Operators and Functions (cont.)



❖ Location-path expressions

- Combine node-set operators and functions

- Select last **bold** element node in **head** element node

```
head/title[ last() ]
```

- Select third book element

```
book[ position() = 3 ]
```

– Or alternatively

```
book[ 3 ]
```

- Return total number of element-node children

```
count( * )
```

- Select all book element nodes in document

```
//book
```



```
1  <?xml version = "1.0"?>
2
3  <!-- Fig. 11.13 : stocks.xml -->
4  <!-- Stock list          -->
5
6  <stocks>
7
8      <stock svmbol = "INTC">
9          <name>Intel Corporation</name>
10     </stock>
11
12     <stock svmbol = "CSCO">
13         <name>Cisco Svstems. Inc.</name>
14     </stock>
15
16     <stock svmbol = "DELL">
17         <name>Dell Computer Corporation</name>
18     </stock>
19
20     <stock svmbol = "MSFT">
21         <name>Microsoft Corporation</name>
22     </stock>
23
24     <stock svmbol = "SUNW">
25         <name>Sun Microsvstems. Inc.</name>
26     </stock>
27
28     <stock svmbol = "CMGI">
29         <name>CMGI. Inc.</name>
30     </stock>
31
32 </stocks>
```



```
1  <?xml version = "1.0"?>
2
3  <!-- Fig. 11.14 : stocks.xsl -->
4  <!-- string function usage -->
5
6  <xsl:stylesheet version = "1.0"
7    xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
8
9    <xsl:template match = "/stocks">
10      <html>
11        <body>
12          <ul>
13
14            <xsl:for-each select = "stock">
15
16              <xsl:if test =
17                "starts-with(@symbol, 'C')">
18
19                <li>
20                  <xsl:value-of select =
21                    "concat(@symbol, ' - ', name)"/>
22                </li>
23              </xsl:if>
24
25            </xsl:for-each>
26          </ul>
27        </body>
28      </html>
29    </xsl:template>
30  </xsl:stylesheet>
```

XPath string functions

Reference



- ❖ XML How to program
- ❖ <http://www.w3.org>
- ❖ XML tutorial
<http://www.w3schools.com/w3c/>



- ❖ Feel free to post questions at <http://yht4ever.blogspot.com>
- ❖ or email to: thanh.phamhong@niithoasen.com or thanh.phamhong@niit-vn.com



Thank You !

<http://yht4ever.blogspot.com>