From: www.luckstar.com.tw

Date: 2013-03-03

Subject: XPath語法

XPath uses path expressions to select nodes or node-sets in an XML document. The node is selected by following a path or steps.

## The XML Example Document

We will use the following XML document in the examples below.

< ?xml version="1.0" encoding="ISO-8859-1"?>  
  
< bookstore>  
  
< book>  
< title lang="eng">Harry Potter</title>  
< price>29.99</price>  
< /book>  
  
< book>  
< title lang="eng">Learning XML</title>  
< price>39.95</price>  
< /book>  
  
< /bookstore>

## Selecting Nodes

XPath uses path expressions to select nodes in an XML document. The node is selected by following a path or steps. The most useful path expressions are listed below:

|  |  |
| --- | --- |
| **Expression** | **Description** |
| *nodename* | Selects all nodes with the name "*nodename*" |
| / | Selects from the root node |
| // | Selects nodes in the document from the current node that match the selection no matter where they are |
| . | Selects the current node |
| .. | Selects the parent of the current node |
| @ | Selects attributes |

In the table below we have listed some path expressions and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| bookstore | Selects all nodes with the name "bookstore" |
| /bookstore | Selects the root element bookstore  **Note:** If the path starts with a slash ( / ) it always represents an absolute path to an element! |
| bookstore/book | Selects all book elements that are children of bookstore |
| //book | Selects all book elements no matter where they are in the document |
| bookstore//book | Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element |
| //@lang | Selects all attributes that are named lang |

## Predicates

Predicates are used to find a specific node or a node that contains a specific value.

Predicates are always embedded in square brackets.

In the table below we have listed some path expressions with predicates and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| /bookstore/book[1] | Selects the first book element that is the child of the bookstore element.  **Note:** IE5 and later has implemented that [0] should be the first node, but according to the W3C standard it should have been [1]!! |
| /bookstore/book[last()] | Selects the last book element that is the child of the bookstore element |
| /bookstore/book[last()-1] | Selects the last but one book element that is the child of the bookstore element |
| /bookstore/book[position()<3] | Selects the first two book elements that are children of the bookstore element |
| //title[@lang] | Selects all the title elements that have an attribute named lang |
| //title[@lang='eng'] | Selects all the title elements that have an attribute named lang with a value of 'eng' |
| /bookstore/book[price>35.00] | Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00 |
| /bookstore/book[price>35.00]/title | Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00 |

## Selecting Unknown Nodes

XPath wildcards can be used to select unknown XML elements.

|  |  |
| --- | --- |
| **Wildcard** | **Description** |
| \* | Matches any element node |
| @\* | Matches any attribute node |
| node() | Matches any node of any kind |

In the table below we have listed some path expressions and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| /bookstore/\* | Selects all the child nodes of the bookstore element |
| //\* | Selects all elements in the document |
| //title[@\*] | Selects all title elements which have any attribute |

## Selecting Several Paths

By using the | operator in an XPath expression you can select several paths.

In the table below we have listed some path expressions and the result of the expressions:

|  |  |
| --- | --- |
| **Path Expression** | **Result** |
| //book/title | //book/price | Selects all the title AND price elements of all book elements |
| //title | //price | Selects all the title AND price elements in the document |
| /bookstore/book/title | //price | Selects all the title elements of the book element of the bookstore element AND all the price elements in the document |