

XPath	
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	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

Wildcards and Multiple Paths	
*	Matches any element node
@*	Matches any attribute node
node()	Matches any node of any kind
	Place between paths to select several paths
/books- store/*	Selects all the child element nodes of the bookstore element
//*	Selects all elements in the document
//title[@*]	Selects all title elements which have at least one attribute of any kind
//book/titl e //book/pri ce	Selects all the title AND price elements of all book elements
//title //price	Selects all the title AND price elements in the document

Wildcards and Multiple Paths (cont)	
/book	Selects all the title elements of the stor-
e/b-	book element of the bookstore
ook/titl-	element AND all the price
e	elements in the document
//price	

Location Step Examples	
The syntax for a location step is:	
axisname::no det est [pr edi c ate]	

child::b ook	Selects all book nodes that are children of the current node
attrib- ute::lan- g	Selects the lang attribute of the current node
child::*	Selects all element children of the current node
attrib- ute::*	Selects all attributes of the current node
child::t ext()	Selects all text node children of the current node
child::n ode()	Selects all children of the current node
desce ndan- t::book	Selects all book descendants of the current node
ancest or::boo- k	Selects all book ancestors of the current node
ancest or-or-- sel- f::book	Selects all book ancestors of the current node - and the current node as well if it is a book node
child::*/ chi- ld::pric- e	Selects all price grandchildren of the current node

XPath Operators		
	Computes two node-sets	//book //cd
+	Addition	6 + 4
-	Subtraction	6 - 4
*	Multiplication	6 * 4
div	Division	8 div 4
=	Equal	price=9.80
!=	Not equal	price!=9.80
<	Less than	price<9.80
<=	Less than or equal to	price<=9.80
>	Greater than	price>9.80
>=	Greater than or equal to	price>=9.80
or	or	price=9.80 or price=9.70
and	and	price>9.00 and price<9.90
mod	Modulus	5 mod 2

Predicates	
/books- store/b- ook[1]	Selects the first book element that is the child of the bookstore element.
/books- store/b- ook[la- st()]	Selects the last book element that is the child of the bookstore element
/books- store/b- ook[la- st()-1]	Selects the last but one book element that is the child of the bookstore element



Predicates (cont)

`/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='en']` Selects all the title elements that have a "lang" attribute with a value of "en"

`/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

Note: In IE 5,6,7,8,9 first node is [0], but according to W3C, it is [1]. To solve this problem in IE, set the SelectionLanguage to XPath.

In JavaScript: `xml.setProperty("SelectionLanguage","XPath");`

XPath Examples

`bookstore` Selects all nodes with the name "bookstore"

`/bookstore` Selects the root element bookstore

`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

`/bookstore/book/title` Selects all title nodes that are descendants of book that are descendants of bookstore

`/bookstore/book[1]/title` Selects the title of the first book node under the bookstore element

`/bookstore/book/price[text()]` Selects the text from all bookstore/book/price nodes

`/bookstore/book[price > 35]/price` Selects all the bookstore/book/price nodes with a price greater than 35

Note: If the path started with a slash (/) it always represents an absolute path to an element!

XPath Axes

`ancestor` Selects all ancestors of the current node

`ancestor-or-self` Selects all ancestors of the current node and the current node itself

`attribute` Selects all attributes of the current node

`child` Selects all children of the current node

`descendant` Selects all descendants of the current node

`descendant-or-self` Selects all descendants of the current node and the current node itself

`following` Selects everything in the document after the closing tag of the current node

`following-sibling` Selects all siblings after the current node

`namespace` Selects all namespace nodes of the current node

`parent` Selects the parent of the current node

`preceding` Selects all nodes that appear before the current node in the document, except ancestors, attribute nodes, and namespace nodes

`preceding-sibling` Selects all siblings before the current node

`self` Selects the current node



By **BenHuf**
cheatography.com/benhuf/

Not published yet.
Last updated 25th September, 2022.
Page 2 of 2.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopad.com>