The XML Path Language (XPath)

The XML Path Language (XPath) is a query language used to identify and select XML nodes (i.e., elements, attributes, etc.) in an XML or XML-like document (like an HTML source of a web page) using a "path-like" expression. It allows you to identify nodes in an XML document in cases where they don't have a unique tag ID or any other identifier.

Syntax: *node_selector*[*predicate*]...

Node Selectors:

Node selector prefixes:	
/	- Indicates that the following node selector is applied only to the direct children of the current node.
//	- Indicates that the following node selector is applied to every descendant of the current node. (Please note that every node can be selected only once, so if a particular node is a descendant of multiple nodes, it will not be selected multiple times, but only once).

. - Selects the current node.

.. - Selects the parent of the current node.

element - Selects element nodes with the name '*element*'.

@attribute - Selects attribute nodes with the name '*attribute*'.

Predicates:

- The predicate is an expression used to filter the found nodes. The predicate expression is checked for each node that was found by the node selector. If the predicate expression evaluates to a Boolean value of 'true' (or to a non-empty set) for the current checked node, that node will be selected, otherwise, it will not be selected.
- A predicate is always embedded in square brackets.
- A predicate can contain functions and even other node selectors, which can contain their own predicate. A node selector returns a set of nodes, which can be treated as a Boolean value of 'true' when the set is not empty, and as a Boolean value of 'false' when the set is empty.
- When performing an equality check (using any of the equality operators ['=', '!=', '>', '<', '<=', '>=']) on a set of nodes, the operand will be checked against the value of each node in the set. If at least one node has a value that meets the equality condition, the equality check will return a Boolean value of 'true', otherwise, it will return a Boolean value of 'false'.
- You can surround expressions with parentheses and use logical operators between ('and', 'or').
- When using only the 'position() = n' predicate you can replace it with 'n' ('n' can be any number starting from 1).

Functions:

- Functions return a value based on the current state or on the passed arguments (if there are any).
- The following functions are the most used functions. For a full list of functions please refer to the full XPath specification.

position()	- Returns the index of the current node in the current context.
last()	- Returns the index of the last node in the current context.
text()	- Returns the text of the current checked node in the found nodes collection.
normalize-space(*str*)	- Returns the '*str*' string without any leading or trailing spaces (i.e., space characters, tabs, newlines, etc.) and with sequences of spaces replaced by a single space character.
count(*node_set*)	- Returns the count of nodes in the '*node_set*' node set.
contains(*s1*, *s2*)	- Returns 'true' if the '*s1*' string literal contains the '*s2*' string literal, otherwise returns 'false'.
starts-with(*s1*, *s2*)	- Returns 'true' if the '*s1*' string literal starts with the '*s2*' string literal, otherwise returns 'false'.
ends-with(*s1*, *s2*)	- Returns 'true' if the '*s1*' string literal ends with the '*s2*' string literal, otherwise returns 'false'.
not(*value*)	- Returns 'true' if the '*value*' Boolean value equals to 'false', otherwise returns 'false'.

Examples:

- Please note, the following examples show you only how to select elements from an XML document. An element is one of the types of an XML node. In our case, the XPath is used only for locating elements within a web page (i.e., within the HTML source of a web page), so locating other types of XML nodes is not relevant.
- In the following examples, the starting node is always the document's root node.
- The direct child elements of the document's root node are actually all the top-most elements in the document. When we select, for example, all the "div" elements that are direct children of the document's root node, we are actually selecting all the top-most "div" elements in the document.
- The descendant elements of the document's root node are actually all the elements in the document (excluding the document's root node). When we select, for example, all the "div" elements that are descendants of the document's root node, we are actually selecting all the "div" elements in the document.

/div	- Selects all the "div" elements that are direct children of the document's root node (in our case, all the topmost "div" elements in the document will be selected).
//div	- Selects all the "div" elements that are descendants of the document's root node (in our case, all the "div" elements in the document will be selected, no matter where they are located).

/div/span	- Selects all the "span" elements that are direct children of the "div" elements that are the top-most div elements in the document.
/div//span	- Selects all the "span" elements that are descendants of the "div" elements that are the top-most div elements in the document.
//div/span	- Selects all the "span" elements that are direct children of any "div" element in the document.
//div//span	- Selects all the "span" elements that are descendants of any "div" element in the document.
/*	- Selects all the top-most elements in the document.
//*	- Selects all the elements in the document.
//div/*	- Selects all the elements that are direct children of any "div" element in the document.
//div//*	- Selects all the elements that are descendants of any "div" element in the document.
//div[position() = 3]	- Selects the third "div" element that is found in the document.
//div[3]	- Selects the third "div" element that is found in the document (same as above).
(//div)[position() = 3]	- Selects the third "div" element that is found in the document.
(//div)[3]	- Selects the third "div" element that is found in the document (same as above).
//div[position() <= 3]	- Selects all the "div" elements in the document that are the first up to the third child of their parents.
//div[position() < 3]	- Selects all the "div" elements in the document that are the first up to the second child of their parents.
//div[position() >= 3]	- Selects all the "div" elements in the document that are the third child and up of their parents.
//div[position() > 3]	- Selects all the "div" elements in the document that are the fourth child and up of their parents.
//div[last()]	- Selects all the "div" elements in the document that are the last child of their parents.
//div[last() - 1]	- Selects all the "div" elements in the document that are the one before the last child of their parents.
(//div)[last()]	- Selects the last "div" element that is found in the document.
(//div)[last() - 1]	- Selects the one before the last "div" element that is found in the document.
//div[@width]	- Selects all the "div" elements in the document that have a "width" attribute.
//div[not(@width)]	- Selects all the "div" elements in the document that don't have a "width" attribute.

//div[@width = 100]	- Selects all the "div" elements in the document that have a "width" attribute and its numerical value equals to 100.
//div[@width != 100]	- Selects all the "div" elements in the document that have a "width" attribute and its numerical value doesn't equal to 100.
//div[@width > 100]	- Selects all the "div" elements in the document that have a "width" attribute and its numerical value is above 100.
//div[span]	- Selects all the "div" elements in the document that have a "span" child element.
//div[not(span)]	- Selects all the "div" elements in the document that don't have a "span" child element.
//div[span = 100]	- Selects all the "div" elements in the document that have a "span" child element and its numerical value equals to 100.
//div[span != 100]	- Selects all the "div" elements in the document that have a "span" child element and its numerical value doesn't equal to 100.
//div[span > 100]	- Selects all the "div" elements in the document that have a "span" child element and its numerical value is above 100.
//div[text() = "Test"]	- Selects all the "div" elements in the document that have exactly the text "Test".
//div[@name = "Test"]	- Selects all the "div" elements in the document that have a "name" attribute and it has exactly the text "Test".
//div[span = "Test"]	- Selects all the "div" elements in the document that have a "span" child element and it has exactly the text "Test".
//div[contains(text(), "Test")]	- Selects all the "div" elements in the document that contain the text "Test".
//div[contains(@name, "Test")]	- Selects all the "div" elements in the document that have a "name" attribute and it contains the text "Test".
//div[contains(span, "Test")]	- Selects all the "div" elements in the document that have a "span" child element and it contains the text "Test".
//div[starts-with(text(), "Test")]	- Selects all the "div" elements in the document that their text starts with "Test".
//div[ends-with(text(), "Test")]	- Selects all the "div" elements in the document that their text ends with "Test".
//div[normalize-spaces(text()) = "Test"]	- Selects all the "div" elements in the document that have exactly the text "Test", after removing any leading or trailing spaces (i.e., space characters, tabs, newlines, etc.) and replacing sequences of spaces by a single space character.

//div[count(span) = 3]	- Selects all the "div" elements in the document that have exactly 3 "span" child nodes.
//div[(@width = 100) and (@height = 200)]	- Selects all the "div" elements in the document that have a "width" attribute with a numerical value of 100, and a "height" attribute with a numerical value of 200.
//div[(@width = 100) or (@height = 200)]	- Selects all the "div" elements in the document that have a "width" attribute with a numerical value of 100, or a "height" attribute with a numerical value of 200.
//div[@width = 100]/span[@width = 200]	- Selects all the "span" elements that have a "width" attribute with a numerical value of 200 and are direct children of any "div" element in the document that have a "width" attribute with a numerical value of 100.
//div[@width = 100]//span[@width = 200]	- Selects all the "span" elements that have a "width" attribute with a numerical value of 200 and are descendants of any "div" element in the document that have a "width" attribute with a numerical value of 100.
//div[span[@width = 100]]	- Selects all the "div" elements in the document that have at least one "span" child element that has a "width" attribute and its numerical value is 100.

Disclaimer:

This document is only a short summary of the most common aspects of the XML Path Language (XPath).

The full XPath specification can be found on: https://www.w3.org/TR/1999/REC-xpath-19991116/.