Axes

Forward Axes

child::	Direct children of the context node *This is the default axis*
self::	The current node
descendant::	Descendants (children, grandchildren, etc.) of the current node
descendant-or-self::	Descendants (children, grandchildren, etc.) of the current node and the current node itself
following::	All nodes that are after the context node in document order, excluding any ancestors, attribute nodes and namespace nodes
following-sibling::	All siblings that are after the context node in document order
namespace::	Namespace nodes of the context node
attribute::	Attributes of the context node

Reverse Axes

parent::	The direct parent of the context node
ancestor::	All the ancestors of the context node
ancestor-or-self::	Ancestors of the context node and the context node itself
preceding-sibling::	All siblings of the context node that came before the context node
preceding::	All nodes that are before the context node in document order, excluding any ancestors, attribute nodes and namespace nodes

Forward axes only contain nodes that come after the context node in document order

Reverse axes only contain nodes that come before the context node in document order

Node Types

Root

Does not occur except as the root of the tree. The element node for the document element is a child of the root node. The root node also has as children processing instruction and comment nodes for processing instructions and comments that occur in the prolog and after the end of the document element. The string-value of the root node is the concatenation of the string-values of all text node descendants of the root node in document order. The root node does not have an expanded-name.

Element

There is an element node for every element in the document. An element node has an expanded-name computed by expanding the QName of the element specified in the tag in accordance with the XML Namespaces Recommendation [XML Names]. The namespace URI of the element's expanded-name will be null if the QName has no prefix and there is no applicable default namespace. NOTE: In the notation of Appendix A.3 of [XML Names], the local part of the expanded-name corresponds to the type attribute of the ExpEType element; the namespace URI of the expanded-name corresponds to the ns attribute of the ExpEType element, and is null if the ns attribute of the ExpEType element is omitted. The children of an element node are the element nodes, comment nodes, processing instruction nodes and text nodes for its content. Entity references to both internal and external entities are expanded. Character references are resolved. The string-value of an element node is the concatenation of the string-values of all text node descendants of the element node in document order. Its string-value is the concatenation of its children's string values and text node values, recursively.

Node Types (cont)

Attribute

Each element node has an associated set of attribute nodes; the element is the parent of each of these attribute nodes; however, an attribute node is not a child of its parent element. NOTE: This is different from the DOM, which does not treat the element bearing an attribute as the parent of the attribute (see [DOM]). Elements never share attribute nodes: if one element node is not the same node as another element node, then none of the attribute nodes of the one element node will be the same node as the attribute nodes of another element node. NOTE: The = operator tests whether two nodes have the same value, not whether they are the same node. Thus attributes of two different elements may compare as equal using =, even though they are not the same node. A defaulted attribute is treated the same as a specified attribute. If an attribute was declared for the element type in the DTD, but the default was declared as #IMPLIED, and the attribute was not specified on the element, then the element's attribute set does not contain a node for the attribute. Some attributes, such as xml:lang and xml:space, have the semantics that they apply to all elements that are descendants of the element bearing the attribute, unless overridden with an instance of the same attribute on another descendant element. However, this does not affect where attribute nodes appear in the tree: an element has attribute nodes only for attributes that were explicitly specified in the start-tag or empty-element tag of that element or that were explicitly declared in the DTD with a default value. An attribute node has an expanded-name and a string-value. The expanded-name is computed by expanding the QName specified in the tag in the XML document in accordance with the XML Namespaces Recommendation [XML Names]. The namespace URI of the attribute's name will be null if the QName of the attribute does not have a prefix. NOTE: In the notation of Appendix A.3 of [XML Names], the local part of the expanded-name corresponds to the name attribute of the ExpAName element; the namespace URI of the expanded-name corresponds to the ns attribute of the ExpAName element, and is null if the ns attribute of the ExpAName element is omitted. An attribute node has a string-value. The string-value is the normalized value as specified by the XML Recommendation [XML]. An attribute whose normalized value is a zero-length string is not treated specially: it results in an attribute node whose string-value is a zero-length string. NOTE: It is possible for default attributes to be declared in an external DTD or an external parameter entity. The XML Recommendation does not require an XML processor to read an external DTD or an external parameter unless it is validating. A stylesheet or other facility that assumes that the XPath tree contains default attribute values declared in an external DTD or parameter entity may not work with some non-validating XML processors. There are no attribute nodes corresponding to attributes that declare namespaces (see [XML Names]).

Node Types (cont)

Namespace

Each element has an associated set of namespace nodes, one for each distinct namespace prefix that is in scope for the element (including the xml prefix, which is implicitly declared by the XML Namespaces Recommendation [XML Names]) and one for the default namespace if one is in scope for the element. The element is the parent of each of these namespace nodes; however, a namespace node is not a child of its parent element. Elements never share namespace nodes: if one element node is not the same node as another element node, then none of the namespace nodes of the one element node will be the same node as the namespace nodes of another element node. This means that an element will have a namespace node: for every attribute on the element whose name starts with xmlns:; for every attribute on an ancestor element whose name starts xmlns: unless the element itself or a nearer ancestor redeclares the prefix; for an xmlns attribute, if the element or some ancestor has an xmlns attribute, and the value of the xmlns attribute for the nearest such element is non-empty NOTE: An attribute xmlns="" "undeclares" the default namespace (see [XML Names]). A namespace node has an expanded-name: the local part is the namespace prefix (this is empty if the namespace node is for the default namespace); the namespace URI is always null. The string-value of a namespace node is the namespace URI that is being bound to the namespace prefix; if it is relative, it must be resolved just like a namespace URI in an expanded-name.

For the attribute axis, the principal node type is attribute.
For the namespace axis, the principal node type is namespace.
For other axes, the principal node type is element.

By **ShaunWinters**
cheatography.com/shaunwinters/

Not published yet.
Last updated 10th May, 2016.
Page 2 of 2.

Sponsored by **Readability-Score.com**
Measure your website readability!
https://readability-score.com