XPath 103 Cheat Sheet: Axes (node relations)

XPath axes are used as follow: `node/axis::predicate`

`ancestor`: Selects all the parents and grand parents of the current node till the root node. For example, `//div[@class="header"]/ancestor::div` will select each and every `div` node that is a parent or a grandparent to the current `div` node.

`parent`: Selects the direct parent of the current node only. For example, `//li/parent::ul` will match the `ul` element that is parent to this `li` .

`child`: Selects the direct child of the current node only. For example, `//ul/child::li` will select the `li` nodes that are direct children to the current `ul` node. Notice that the same can be achieved with the `/` symbol as follows `//ul/li`.

`descendant`: Selects all the children and grandchildren of the current node regardless of whether they are direct children. For example, `/html/descendant::ul`, will select all the `ul` elements in the document although they are not direct children to `/html`, which is the root node. Notice that the same can be achieved using `//ul`.

`following`: Selects all the nodes that comes after the closing tag of the current tag regardless of their relation. For example, `//div/following::span` will match all `span` elements that comes after this `div` element even if not children or grandchildren.

`following-sibling`: Like the `following` axis but the selection is limited to the direct sibling of the current node. For example, `//div/following::span` will match all `span` nodes that come after the closing tag of the current `div` node only if they share the same parent node.

`preceding`: Selects all the nodes that come before the current node except ancestors, attribute nodes or namespace nodes. For example, `//li/preceding::ul` will **not match** anything since the `ul` is a parent for `li`. But `//li/preceding::div` will match any `div` node preceding the `li` node as long as it is *not* one of its ancestors.

`preceding-sibling`: Like `preceding` but it will only match nodes that share the same parent as the current node. For example, `//li/preceding::div` will probably match nothing unless the `ul` node has a child `div` at the same level as the `li` node.