XPath - Reference

Absolute XPath:

Absolute XPath begins from the root element "/" and traverses through the DOM tree until it reaches the desired element. Note that if the DOM tree structure above the desired element changes in future, the absolute path will not work.

Example:`/html[1]/body[1]/div[1]/div[4]/div[1]/div[2]/div[1]/div[1]/div[2]/div[2]/div[2]/ul[1]/li[1]`

Begins at the root element / and traverses until it reaches the desired element `li`

RelativeXPath:

Relative XPath originates anywhere in the DOM tree beginning with whichever element you want using "//".

Example:
```
//a[@id="gb78"]
//a[@id="gb78"]/span
```

Begins at the specified element `a[@id="gb78"]`

XPath Functions:

starts-with()

The `starts-with` checks whether the value of attribute(first argument) starts with the given value(second argument) and returns true or false.

Example:
```
//a[starts-with(@href, "https://play.google.com/")]
//*[starts-with(@href, "https://play.google.com/")]
```

Checks if the `href` attribute value starts with "`https://play.google.com/`" and returns true or false.

contains()

The `contains` checks whether the value of attribute(first argument) contains anywhere the given value(second argument) and returns true or false.

Example: `//a[contains(@href, "play.google.com/")]`

Checks if the `href` attribute value contains "`https://play.google.com/`" and returns true or false.

text()

The `text` checks whether the inner text matches the given value and returns true or false.

Example: `//span[text()="Play"]`

Checks if the `span` element has inner text that matches "`Play`" and returns true or false.

XPath Axes:

preceding-sibling

Indicates all the nodes that have the same parent as the context node and appear before the context node in the source document.

Example: `//span[text() = "Play"]/preceding-sibling::span`

Locates the `span` element that precedes the element `//span[text() = "Play"]` and have the same parent.

following-sibling

Indicates all the nodes that have the same parent as the context node and appear after the context node in the source document.

Example: `//a[text()="More"]/following-sibling::a`

Locates the `a` element that follows the element `//a[text()="More"]` and have the same parent.

parent

Indicates the single node that is the parent of the context node. It can be abbreviated as two periods (`..`).

Example: `//a[starts-with(@href, "https://play.google.com/")]/parent::li`

`//a[starts-with(@href, "https://play.google.com/")]/..`

Locates the parent for `//a[starts-with(@href, "https://play.google.com/")]`.

ancestor

Indicates the nodes that are ancestors(parent, grandparent and more) of the context node.

Example: `//a[starts-with(@href, "https://play.google.com")]/ancestor::ul`

Locates the ancestor `ul` element for "`//a[starts-with(@href, "https://play.google.com/")]`".

child

Indicates the direct children of the context node.

Example: `//a[starts-with(@href, "https://play.google.com/")]/child::span`

Locates the direct children `span` for `//a[starts-with(@href, "https://play.google.com/")]`.

descendant

Indicates all the descendants(children, grandchildren and more) of the context node.

Example: `//a[starts-with(@href, "https://play.google.com/")]/descendant::span`

Locates all the descendant `span` for `//a[starts-with(@href, "https://play.google.com/")]`.

For Further Reading: https://developer.mozilla.org/en-US/docs/Web/XPath