

XPath 102 Cheat Sheet

`/`: Used to select the root node or the immediate child node. For example, `/html` will select the root node, `html` and all its descendants (the entire document). If the query is `div/p` then we are selecting the first node of type `p` (paragraph) that comes immediately after the selected `div`.

`//`: Used to search for the node regardless of its parent. When using `//`, we needn't start our path from the root node. Instead, we can *jump* to our node. For example, `//div` will select the first node of type `div` in the document. It can also be used in the middle of a query. For example, `//div//a[@id="98760"]` will match the first node of type `a` (hyperlink) in the *descendants* of `div`. It will ignore any other `div` child nodes until it find the `a` node with an `id` attribute that equals `98760`.

`@`: Used to select a node attribute. It is commonly used in conjunction with `//` to narrow down the selection. For example, while `//div` will match the first `div` node in the document, `//div[@class="title"]` will match the first `div` node that has a class of `title` i.e. `<div class="title">some content</div>`

`/text()`: When added to the end of a query, extracts the text part of the node. For example, applying `//div[@class="title"]/text()` against `<div class="title">some content</div>` will return `some content`.

`//@attribute node`: Will match all the nodes that has an attribute node (regardless of the value of that attribute). For example, `//@data-type` will match any HTML element that has a `data-type` attribute. But it doesn't care about the value of this `data-type`.

`node [n]`: Will match the `n`th node in a matched list. For example, `//li[2]` will match the second `li` node in the list of matched `li` nodes.

`node [last()]`: Will match the last node in a matched list. For example, `li[last()]` will match the last `li` node in the list of matched `li` nodes.

`node [last()-n]`: Will match the `n` before last item in a list of matched nodes. For example, in a `li` list of ten items, `li[last()-2]` will match the eighth `li`.

`node [position()<n]`: In a list of nodes, it will match the first number of nodes that is less than `n`. For example, `li[position()<2]` will match the first node (node number 1 since it is less than 2). `li[position()<10]` will match the *first nine nodes* in the list.

`*`: Is a greedy match. It will select all nodes. Combined with selectors, it will match all the nodes that satisfy the query regardless of their type. For example, `//*[@class="col-md-4"]` will match any and all the nodes that have the attribute `class="col-md-4"`.

`|`: combines queries together with an **AND** operator. For example, `//h1[@class="header"] | h2[@class="header"]` will return all `h1` and `h2` headers that have the attribute `class="header"`.