

LXML Element Reference

NAVIGATING ELEMENTS

`body` HTML
`cssselect(expr, translator='html')` HTML
`find(path, namespaces=None)`
`find_class(class_name)` HTML
`find_rel_links(rel)` HTML
`findall(path, namespaces=None)`
`forms` HTML
`get_element_by_id(id[, default])` HTML
`getnext()`
`getparent()`
`getprevious()`
`getroottree()`
`head` HTML
`index(child, start=None, stop=None)`
`iter(tag=None)`
`iterancestors(tag=None)`
`iterchildren(tag=None, reversed=False)`
`iterdescendants(tag=None)`
`iterfind(path, namespaces=None)`
`iterlinks()` HTML
`itersiblings(tag=None, preceding=False)`
`label` HTML
`xpath(_path, namespaces=None, extensions=None, smart_strings=True, **_variables)`

LIST BEHAVIOR (CHILD ELEMENTS)

child in *element*
`del element[i]`
for *child* in *element*:
`len(element)`
`reversed(element)`
`element[i] = child`

TEXT EXTRACTION

`findtext(path, default=None, namespaces=None)`
`itertext(tag=None, with_tail=True)`
`tail`
`text`
`text_content()` HTML

GETTING ATTRIBUTES

`attrib`
`get(key, default=None)`
`items()`
`keys()`
`set(key, value)`
`values()`

<FORM>

`action` HTML
`fields` HTML
`inputs` HTML
`method` HTML

<INPUT>

`checkable` HTML
`checked` HTML
`name` HTML
`type` HTML
`value` HTML

<SELECT>

`multiple` HTML
`name` HTML
`value` HTML
`value_options` HTML

<TEXTAREA>

`name` HTML
`value` HTML

<LABEL>

`for_element` HTML

MODIFICATIONS

`addnext(element)`
`addprevious(element)`
`append(element)`
`clear()`
`drop_tag()` HTML
`drop_tree()` HTML
`extend(elements)`
`insert(index, element)`
`makeelement(_tag, attrib=None, nsmap=None, **_extra)`
`make_links_absolute(base_url=None, resolve_base_href=None)` HTML
`remove(element)`
`replace(old_element, new_element)`
`rewrite_links()` HTML
`resolve_base_href()` HTML

SOURCE CONTEXT

`base`
`base_url` HTML
`nsmap`
`prefix`
`sourceline`
`tag`

HTML = HTML only tag=tag name path=XPath expression () gray text=(rarely used)

XPath syntax: <http://effbot.org/zone/element-xpath.htm> lxml API documentation: <http://lxml.de/api/>

XPath Support in ElementTree

syntax	meaning
tag	Selects all child elements with the given tag. For example, “spam” selects all child elements named “spam”, “spam/egg” selects all grandchildren named “egg” in all child elements named “spam”. You can use universal names (“{url}local”) as tags.
*	Selects all child elements. For example, “*/egg” selects all grandchildren named “egg”.
.	Select the current node. This is mostly useful at the beginning of a path, to indicate that it’s a relative path.
//	Selects all subelements, on all levels beneath the current element (search the entire subtree). For example, “./egg” selects all “egg” elements in the entire tree.
..	Selects the parent element.
[@attrib]	Selects all elements that have the given attribute. For example, “./a[@href]” selects all “a” elements in the tree that has a “href” attribute.
[@attrib='value']	Selects all elements for which the given attribute has the given value. For example, “./div[@class='sidebar']” selects all “div” elements in the tree that has the class “sidebar”. In the current release, the value cannot contain quotes.
[tag]	Selects all elements that has a child element named tag. In the current version, only a single tag can be used (i.e. only immediate children are supported).
[position]	Selects all elements that are located at the given position. The position can be either an integer (1 is the first position), the expression “last()” (for the last position), or a position relative to last() (e.g. “last()-1” for the second to last position). This predicate must be preceded by a tag name.

Source: <http://effbot.org/zone/element-xpath.htm>