NAME

XML::LibXML::NodeList − a list of XML document nodes

DESCRIPTION

An XML::LibXML::NodeList object contains an ordered list of nodes, as detailed by the W3C DOM documentation of Node Lists.

SYNOPSIS

```
my $results = $dom->findnodes('//somepath');
foreach my $context ($results->get_nodelist) {
  my $newresults = $context->findnodes('./other/element');
  ...
}
```

API

new(@nodes)

You will almost never have to create a new NodeList object, as it is all done for you by XPath.

get_nodelist()

Returns a list of nodes, the contents of the node list, as a perl list.

string_value()

Returns the string-value of the first node in the list. See the XPath specification for what ''string-value'' means.

to_literal()

Returns the concatenation of all the string-values of all the nodes in the list.

to_literal_delimited($separator)

Returns the concatenation of all the string-values of all the nodes in the list, delimited by the specified separator.

to_literal_list()

Returns all the string-values of all the nodes in the list as a perl list.

get_node($pos)

Returns the node at $pos. The node position in XPath is based at 1, not 0.

size()

Returns the number of nodes in the NodeList.

pop()

Equivalent to perl's pop function.

push(@nodes)

Equivalent to perl's push function.

append($nodelist)

Given a nodelist, appends the list of nodes in $nodelist to the end of the current list.

shift()

Equivalent to perl's shift function.

unshift(@nodes)

Equivalent to perl's unshift function.

prepend($nodelist)

Given a nodelist, prepends the list of nodes in $nodelist to the front of the current list.

map($coderef)

Equivalent to perl's map function.

grep($coderef)

Equivalent to perl's grep function.

sort($coderef)
>
> Equivalent to perl's sort function.
>
> Caveat: Perl's magic $a and $b variables are not available in $coderef. Instead the two terms are passed to the coderef as arguments.

reverse()
>
> Equivalent to perl's reverse function.

foreach($coderef)
>
> Inspired by perl's foreach loop. Executes the coderef on each item in the list. Similar to map, but instead of returning the list of values returned by $coderef, returns the original NodeList.

reduce($coderef, $init**)**
>
> Equivalent to List::Util's reduce function. $init is optional and provides an initial value for the reduction.
>
> Caveat: Perl's magic $a and $b variables are not available in $coderef. Instead the two terms are passed to the coderef as arguments.