
NAME

XML::LibXML::Document - DOM Document Class

SYNOPSIS

```
  use XML::LibXML::Document;

  $dom = XML::LibXML::Document->new( $version, $encoding );
  $dom = XML::LibXML::Document->createDocument( $version, $encoding );
  $strEncoding = $doc->getEncoding();
  $strVersion = $doc->getVersion();
  $docstring = $dom->toString([$format]);
  $bool = $dom->is_valid();
  $root = $dom->getDocumentElement($name, $namespace );
  $dom->setDocumentElement( $root );
  $element = $dom->createElement( $nodename );
  $element = $dom->createElementNS( $namespaceURI, $qname );
  $text = $dom->createTextNode( $content_text );
  $comment = $dom->createComment( $comment_text );
  $attrnode = $doc->createAttribute($name [,$value]);
  $attrnode = $doc->createAttributeNS( namespaceURI, $name [,$value] );
  $cdata = $dom->create( $cdata_content );
  $document->importNode( $node [, $move] );
```

DESCRIPTION

The Document Class is the result of a parsing process. But sometimes it is necessary to create a Document from scratch. The DOM Document Class provides functions that are conform to the DOM Core naming style. It inherits all functions from *XML::LibXML::Node* as specified in DOM Level2. This enables to access the nodes beside the root element on document level - a *DTD* for example. The support for these nodes is limited at the moment, so I would recommend, not to use *node* functions on *documents*. It is suggested that one should always create a

node not bound to any document. There is no need of really including the node to the document, but once the node is bound to a document, it is quite safe that all strings have the correct encoding. If an unbound textnode with an iso encoded string is created (e.g. with $CLASS->new()), the *toString* function may not return the expected result. This seems like a limitation as long UTF8 encoding is assured. If iso encoded strings come into play it is much safer to use the node creation functions of XML::LibXML::Document.

Methods

`new`

alias for `createDocument()`

`createDocument`

The constructor for the document class. As Parameter it takes the version string and (optionally) the ecoding string. Simply calling `createDocument` will create the document:

```
<?xml version="your version" encoding="your encoding"?>
```

Both parameter are optional. The default value for `$version` is *1.0* , of course. If the `$encoding` parameter is not set, the encoding will be left unset, which means UTF8 is implied (and set). The call of `createDocument` without any parameter will result the following code:

```
<?xml version="1.0"?>
```

`getEncoding`

returns the encoding string of the document

`getVersion`

returns the version string of the document

`toString`

`toString` is a deparsing function, so the DOM Tree can be translated into a string, ready for output. The optional `$format` parameter sets the indenting of the output. This parameter is expected to be an *integer* value, that specifies the number of linebreaks for each node. For more information about the formatted output check the documentation of *xmlDocDumpFormatMemory* in *libxml2/tree.h* .

is_valid

Returns either TRUE or FALSE depending on the DOM Tree is a valid Document or not.

getDocumentElement

Returns the root element of the Document. A document can have just one root element to contain the documents data.

setDocumentElement

This function enables you to set the root element for a document. The function supports the import of a node from a different document tree.

createElement

This function creates a new Element Node bound to the DOM with the name *$nodename* .

createElementNS

This function creates a new Element Node bound to the DOM with the name *$nodename* and placed in the given namespace.

createTextNode

As an equivalent of createElement , but it creates a Text Node bound to the DOM.

createComment

As an equivalent of createElement , but it creates a Comment Node bound to the DOM.

createAttribute

Creates a new Attribute node. This function is rather useless at the moment, since there is no setAttributeNode function defined in *XML::LibXML::Element* , yet.

createAttributeNS

Creates an Attribute bound to a namespace.

createCDATASection

Similar to createTextNode and createComment, this function creates a CDataSection bound to the current DOM.

importNode

If a node is not part of a document, it can be imported to another document. As specified in DOM Level 2 Specification the Node will not be altered or removed from its original document by default. (*$node*-`cloneNode(1)`> will get called implicitly). Sometimes it is necessary to *move* a node between documents. In such a case the node will not be copied, but removed from the original document.

SEE ALSO

XML::LibXML, XML::LibXML::Element, XML::LibXML::Text, XML::LibXML::Attr, XML::LibXML::Comment

---

VERSION

0.90_a