:mod:`xml.dom.minidom` --- Minimal DOM implementation

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 4)

Unknown directive type "module".

```
.. module:: xml.dom.minidom
   :synopsis: Minimal Document Object Model (DOM) implementation.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 7)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Paul Prescod <paul@prescod.net>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 8)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Paul Prescod <paul@prescod.net>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 9)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Martin v. LÃ¶wis <martin@v.loewis.de>
```

Source code: :source:`Lib/xml/dom/minidom.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 11); *backlink*

Unknown interpreted text role "source".

:mod:`xml.dom.minidom` is a minimal implementation of the Document Object Model interface, with an API similar to that in other languages. It is intended to be simpler than the full DOM and also significantly smaller. Users who are not already proficient with the DOM should consider using the :mod:`xml.etree.ElementTree` module for their XML processing instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 15); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 15); *backlink*

Unknown interpreted text role "mod".

Warning

The :mod:`xml.dom.minidom` module is not secure against maliciously constructed data. If you need to parse

untrusted or unauthenticated data see :ref:`xml-vulnerabilities`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 24); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 24); *backlink*

Unknown interpreted text role "ref".

DOM applications typically start by parsing some XML into a DOM. With :mod:`xml.dom.minidom`, this is done through the parse functions:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 29); *backlink*

Unknown interpreted text role "mod".

```
from xml.dom.minidom import parse, parseString

dom1 = parse('c:\\temp\\mydata.xml')  # parse an XML file by name

datasource = open('c:\\temp\\mydata.xml')
dom2 = parse(datasource)  # parse an open file

dom3 = parseString('<myxml>Some data<empty/> some more data</myxml>')
```

The :func:`parse` function can take either a filename or an open file object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 41); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 44)

Unknown directive type "function".

```
.. function:: parse(filename_or_file, parser=None, bufsize=None)

   Return a :class:`Document` from the given input. *filename_or_file* may be
   either a file name, or a file-like object. *parser*, if given, must be a SAX2
   parser object. This function will change the document handler of the parser and
   activate namespace support; other parser configuration (like setting an entity
   resolver) must have been done in advance.
```

If you have XML in a string, you can use the :func:`parseString` function instead:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 52); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 56)

Unknown directive type "function".

```
.. function:: parseString(string, parser=None)

   Return a :class:`Document` that represents the *string*. This method creates an
   :class:`io.StringIO` object for the string and passes that on to :func:`parse`.
```

Both functions return a :class:`Document` object representing the content of the document.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 61); backlink

Unknown interpreted text role "class".

What the :func:`parse` and :func:`parseString` functions do is connect an XML parser with a "DOM builder" that can accept parse events from any SAX parser and convert them into a DOM tree. The name of the functions are perhaps misleading, but are easy to grasp when learning the interfaces. The parsing of the document will be completed before these functions return; it's simply that these functions do not provide a parser implementation themselves.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 64); backlink

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 64); backlink

Unknown interpreted text role "func".

You can also create a :class:`Document` by calling a method on a "DOM Implementation" object. You can get this object either by calling the :func:`getDOMImplementation` function in the :mod:`xml.dom` package or the :mod:`xml.dom.minidom` module. Once you have a :class:`Document`, you can add child nodes to it to populate the DOM:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 71); backlink

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 71); backlink

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 71); backlink

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 71); backlink

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 71); backlink

Unknown interpreted text role "class".

```
from xml.dom.minidom import getDOMImplementation

impl = getDOMImplementation()

newdoc = impl.createDocument(None, "some_tag", None)
top_element = newdoc.documentElement
text = newdoc.createTextNode('Some textual content.')
top_element.appendChild(text)
```

Once you have a DOM document object, you can access the parts of your XML document through its properties and methods. These properties are defined in the DOM specification. The main property of the document object is the :attr:`documentElement` property. It gives you the main element in the XML document: the one that holds all others. Here is an example program:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 86); backlink

Unknown interpreted text role "attr".

```
dom3 = parseString("<myxml>Some data</myxml>")
assert dom3.documentElement.tagName == "myxml"
```

When you are finished with a DOM tree, you may optionally call the :meth:`unlink` method to encourage early cleanup of the now-unneeded objects. :meth:`unlink` is an :mod:`xml.dom.minidom`-specific extension to the DOM API that renders the node and its descendants are essentially useless. Otherwise, Python's garbage collector will eventually take care of the objects in the tree.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 95); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 95); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 95); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 102)

Unknown directive type "seealso".

```
.. seealso::

   `Document Object Model (DOM) Level 1 Specification <https://www.w3.org/TR/REC-DOM-Level-1/>`_
      The W3C recommendation for the DOM supported by :mod:`xml.dom.minidom`.
```

DOM Objects

The definition of the DOM API for Python is given as part of the :mod:`xml.dom` module documentation. This section lists the differences between the API and :mod:`xml.dom.minidom`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 113); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 113); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 118)

Unknown directive type "method".

```
.. method:: Node.unlink()

   Break internal references within the DOM so that it will be garbage collected on
   versions of Python without cyclic GC.  Even when cyclic GC is available, using
   this can make large amounts of memory available sooner, so calling this on DOM
   objects as soon as they are no longer needed is good practice.  This only needs
   to be called on the :class:`Document` object, but may be called on child nodes
   to discard children of that node.

   You can avoid calling this method explicitly by using the :keyword:`with`
   statement. The following code will automatically unlink *dom* when the
   :keyword:`!with` block is exited::

      with xml.dom.minidom.parse(datasource) as dom:
```

```
... # Work with dom.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 135)

Unknown directive type "method".

```
.. method:: Node.writexml(writer, indent="", addindent="", newl="", \
                          encoding=None, standalone=None)

   Write XML to the writer object.  The writer receives texts but not bytes as input,
   it should have a :meth:`write` method which matches that of the file object
   interface.  The *indent* parameter is the indentation of the current node.
   The *addindent* parameter is the incremental indentation to use for subnodes
   of the current one.  The *newl* parameter specifies the string to use to
   terminate newlines.

   For the :class:`Document` node, an additional keyword argument *encoding* can
   be used to specify the encoding field of the XML header.

   Similarly, explicitly stating the *standalone* argument causes the
   standalone document declarations to be added to the prologue of the XML
   document.
   If the value is set to `True`, `standalone="yes"` is added,
   otherwise it is set to `"no"`.
   Not stating the argument will omit the declaration from the document.

   .. versionchanged:: 3.8
      The :meth:`writexml` method now preserves the attribute order specified
      by the user.

   .. versionchanged:: 3.9
      The *standalone* parameter was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 162)

Unknown directive type "method".

```
.. method:: Node.toxml(encoding=None, standalone=None)

   Return a string or byte string containing the XML represented by
   the DOM node.

   With an explicit *encoding* [1]_ argument, the result is a byte
   string in the specified encoding.
   With no *encoding* argument, the result is a Unicode string, and the
   XML declaration in the resulting string does not specify an
   encoding. Encoding this string in an encoding other than UTF-8 is
   likely incorrect, since UTF-8 is the default encoding of XML.

   The *standalone* argument behaves exactly as in :meth:`writexml`.

   .. versionchanged:: 3.8
      The :meth:`toxml` method now preserves the attribute order specified
      by the user.

   .. versionchanged:: 3.9
      The *standalone* parameter was added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 183)

Unknown directive type "method".

```
.. method:: Node.toprettyxml(indent="\\t", newl="\\n", encoding=None, \
                             standalone=None)

   Return a pretty-printed version of the document. *indent* specifies the
   indentation string and defaults to a tabulator; *newl* specifies the string
   emitted at the end of each line and defaults to ``\n``.

   The *encoding* argument behaves like the corresponding argument of
   :meth:`toxml`.
```

```
The *standalone* argument behaves exactly as in :meth:`writexml`.

.. versionchanged:: 3.8
   The :meth:`toprettyxml` method now preserves the attribute order specified
   by the user.

.. versionchanged:: 3.9
   The *standalone* parameter was added.
```

DOM Example

This example program is a fairly realistic example of a simple program. In this particular case, we do not take much advantage of the flexibility of the DOM.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 210)

Unknown directive type "literalinclude".

```
.. literalinclude:: ../includes/minidom-example.py
```

minidom and the DOM standard

The :mod:`xml.dom.minidom` module is essentially a DOM 1.0-compatible DOM with some DOM 2 features (primarily namespace features).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 218); *backlink*

Unknown interpreted text role "mod".

Usage of the DOM interface in Python is straight-forward. The following mapping rules apply:

- Interfaces are accessed through instance objects. Applications should not instantiate the classes themselves; they should use the creator functions available on the :class:`Document` object. Derived interfaces support all operations (and attributes) from the base interfaces, plus any new operations.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 224); *backlink*

  Unknown interpreted text role "class".

- Operations are used as methods. Since the DOM uses only :keyword:`in` parameters, the arguments are passed in normal order (from left to right). There are no optional arguments. `void` operations return `None`.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 229); *backlink*

  Unknown interpreted text role "keyword".

- IDL attributes map to instance attributes. For compatibility with the OMG IDL language mapping for Python, an attribute `foo` can also be accessed through accessor methods :meth:`_get_foo` and :meth:`_set_foo`. `readonly` attributes must not be changed; this is not enforced at runtime.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 233); *backlink*

  Unknown interpreted text role "meth".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst,

line 233); *backlink*

Unknown interpreted text role "meth".

- The types `short int`, `unsigned int`, `unsigned long long`, and `boolean` all map to Python integer objects.
- The type `DOMString` maps to Python strings. :mod:`xml.dom.minidom` supports either bytes or strings, but will normally produce strings. Values of type `DOMString` may also be `None` where allowed to have the IDL `null` value by the DOM specification from the W3C.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst`, line 241); *backlink*

Unknown interpreted text role "mod".

- `const` declarations map to variables in their respective scope (e.g. `xml.dom.minidom.Node.PROCESSING_INSTRUCTION_NODE`); they must not be changed.
- `DOMException` is currently not supported in :mod:`xml.dom.minidom`. Instead, :mod:`xml.dom.minidom` uses standard Python exceptions such as :exc:`TypeError` and :exc:`AttributeError`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst`, line 249); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst`, line 249); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst`, line 249); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst`, line 249); *backlink*

Unknown interpreted text role "exc".

- :class:`NodeList` objects are implemented using Python's built-in list type. These objects provide the interface defined in the DOM specification, but with earlier versions of Python they do not support the official API. They are, however, much more "Pythonic" than the interface defined in the W3C recommendations.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst`, line 253); *backlink*

Unknown interpreted text role "class".

The following interfaces have no implementation in :mod:`xml.dom.minidom`:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst`, line 259); *backlink*

Unknown interpreted text role "mod".

- :class:`DOMTimeStamp`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 261); *backlink*

Unknown interpreted text role "class".

- :class:`EntityReference`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]xml.dom.minidom.rst, line 263); *backlink*

Unknown interpreted text role "class".

Most of these reflect information in the XML document that is not of general utility to most DOM users.

Footnotes

[1] The encoding name included in the XML output should conform to the appropriate standards. For example, "UTF-8" is valid, but "UTF8" is not valid in an XML document's declaration, even though Python accepts it as an encoding name. See https://www.w3.org/TR/2006/REC-xml11-20060816/#NT-EncodingDecl and https://www.iana.org/assignments/character-sets/character-sets.xhtml.