:mod:`xml.dom.pulldom` --- Support for building partial DOM trees

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)xml.dom.pulldom.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.pulldom.rst, line 4`)
>
> Unknown directive type "module".
>
> ```
> .. module:: xml.dom.pulldom
>    :synopsis: Support for building partial DOM trees from SAX events.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)xml.dom.pulldom.rst, line 7`)
>
> Unknown directive type "moduleauthor".
>
> ```
> .. moduleauthor:: Paul Prescod <paul@prescod.net>
> ```

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

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)xml.dom.pulldom.rst, line 9`); *backlink*
>
> Unknown interpreted text role "source".

The :mod:`xml.dom.pulldom` module provides a "pull parser" which can also be asked to produce DOM-accessible fragments of the document where necessary. The basic concept involves pulling "events" from a stream of incoming XML and processing them. In contrast to SAX which also employs an event-driven processing model together with callbacks, the user of a pull parser is responsible for explicitly pulling events from the stream, looping over those events until either processing is finished or an error condition occurs.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)xml.dom.pulldom.rst, line 13`); *backlink*
>
> Unknown interpreted text role "mod".

> **Warning**
>
> The :mod:`xml.dom.pulldom` 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.pulldom.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.pulldom.rst, line 24`); *backlink*
> >
> > Unknown interpreted text role "ref".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)xml.dom.pulldom.rst, line 28`)
>
> Unknown directive type "versionchanged".
>
> ```
> .. versionchanged:: 3.7.1
>
>    The SAX parser no longer processes general external entities by default to
>    increase security by default. To enable processing of external entities,
>    pass a custom parser instance in::
> ```

```
from xml.dom.pulldom import parse
from xml.sax import make_parser
from xml.sax.handler import feature_external_ges

parser = make_parser()
parser.setFeature(feature_external_ges, True)
parse(filename, parser=parser)
```

Example:

```
from xml.dom import pulldom

doc = pulldom.parse('sales_items.xml')
for event, node in doc:
    if event == pulldom.START_ELEMENT and node.tagName == 'item':
        if int(node.getAttribute('price')) > 50:
            doc.expandNode(node)
            print(node.toxml())
```

`event` is a constant and can be one of:

- :data:`START_ELEMENT`

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main)(Doc)(library)xml.dom.pulldom.rst,` **line 56);** *backlink*
  >
  > Unknown interpreted text role "data".

- :data:`END_ELEMENT`

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main)(Doc)(library)xml.dom.pulldom.rst,` **line 57);** *backlink*
  >
  > Unknown interpreted text role "data".

- :data:`COMMENT`

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main)(Doc)(library)xml.dom.pulldom.rst,` **line 58);** *backlink*
  >
  > Unknown interpreted text role "data".

- :data:`START_DOCUMENT`

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main)(Doc)(library)xml.dom.pulldom.rst,` **line 59);** *backlink*
  >
  > Unknown interpreted text role "data".

- :data:`END_DOCUMENT`

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main)(Doc)(library)xml.dom.pulldom.rst,` **line 60);** *backlink*
  >
  > Unknown interpreted text role "data".

- :data:`CHARACTERS`

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main)(Doc)(library)xml.dom.pulldom.rst,` **line 61);** *backlink*
  >
  > Unknown interpreted text role "data".

- :data:`PROCESSING_INSTRUCTION`

  > **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main)(Doc)(library)xml.dom.pulldom.rst,` **line 62);** *backlink*
  >
  > Unknown interpreted text role "data".
```

- :data:`IGNORABLE_WHITESPACE`

`node` is an object of type :class:`xml.dom.minidom.Document`, :class:`xml.dom.minidom.Element` or :class:`xml.dom.minidom.Text`.

Since the document is treated as a "flat" stream of events, the document "tree" is implicitly traversed and the desired elements are found regardless of their depth in the tree. In other words, one does not need to consider hierarchical issues such as recursive searching of the document nodes, although if the context of elements were important, one would either need to maintain some context-related state (i.e. remembering where one is in the document at any given point) or to make use of the :func:`DOMEventStream.expandNode` method and switch to DOM-related processing.

Subclass of :class:`xml.sax.handler.ContentHandler`.

Subclass of :class:`xml.sax.handler.ContentHandler`.

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

   Return a :class:`DOMEventStream` from the given input. *stream_or_string* may be
   either a file name, or a file-like object. *parser*, if given, must be an
   :class:`~xml.sax.xmlreader.XMLReader` 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:

Unknown directive type "function".

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

   Return a :class:`DOMEventStream` that represents the (Unicode) *string*.
```

Unknown directive type "data".

```
.. data:: default_bufsize

   Default value for the *bufsize* parameter to :func:`parse`.

   The value of this variable can be changed before calling :func:`parse` and
   the new value will take effect.
```

DOMEventStream Objects

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.11
   Support for :meth:`__getitem__` method has been removed.
```

Unknown directive type "method".

```
.. method:: getEvent()

   Return a tuple containing *event* and the current *node* as
   :class:`xml.dom.minidom.Document` if event equals :data:`START_DOCUMENT`,
   :class:`xml.dom.minidom.Element` if event equals :data:`START_ELEMENT` or
   :data:`END_ELEMENT` or :class:`xml.dom.minidom.Text` if event equals
   :data:`CHARACTERS`.
   The current node does not contain information about its children, unless
   :func:`expandNode` is called.
```

Unknown directive type "method".

```
.. method:: expandNode(node)

   Expands all children of *node* into *node*. Example::

       from xml.dom import pulldom

       xml = '<html><title>Foo</title> <p>Some text <div>and more</div></p> </html>'
       doc = pulldom.parseString(xml)
       for event, node in doc:
           if event == pulldom.START_ELEMENT and node.tagName == 'p':
               # Following statement only prints '<p/>'
               print(node.toxml())
               doc.expandNode(node)
               # Following statement prints node with all its children '<p>Some text <div>and more</div
               print(node.toxml())
```

Unknown directive type "method".

```
.. method:: DOMEventStream.reset()
```