

**DOM XML:** Es una implementación mínima de la interfaz de modelado de objetos del documento, está destinada a ser más simple que una implementación completa de DOM y también de un tamaño significativamente más pequeño. Es una interfaz de programación de aplicaciones para documentos HTML y XML. Define la estructura lógica de los documentos y el modo en el que acceden y manipulan el documento. Con estas funcionalidades, se pueden construir documentos, navegar por estructura y añadir, modificar o eliminar elementos y contenidos de todo el documento. En esencia es una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos y una interfaz estándar para poder acceder a ellos y para manipularlos.

```
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>')
```

Crear un fichero XML

```
# coding: utf-8

from xml.dom import minidom

Ordenador1 = ['Pentium M', '512MB']
Ordenador2 = ['Pentium Core 2', '1024MB']
Ordenador3 = ['Pentium Core Duo', '1024MB']
listaOrdenadores = [Ordenador1, Ordenador2, Ordenador3]

# Abro un modelo DOM en modo implementar
DOMimpl = minidom.getDOMImplementation()

# Crear el documento econ la etiqueta principal estacionesTrabajo
xmldoc = DOMimpl.createDocument(None, "estacionesTrabajo", None)
doc_root = xmldoc.documentElement
```

**xPath XML:** XPath es un lenguaje que permite seleccionar nodos de un documento XML y calcular los valores a partir de su contenido. Permite construir expresiones que recorren y procesan un documento XML. Fue creada para seleccionar partes del texto sin atributos. En general se utiliza para navegar y consultar los elementos y atributos contenidos en la estructura de un documento XML. Sirve para definir criterios de búsqueda avanzada y cálculos específicos utilizando una sintaxis simple.

```
from elementtree.ElementTree import ElementTree
mydoc = ElementTree(file='tst.xml')
for e in mydoc.findall('/foo/bar'):
    print e.get('title').text
```

```
from xml.dom.ext.reader import Sax2
from xml import xpath
doc = Sax2.FromXmlFile('foo.xml').documentElement
for url in xpath.Evaluate('//@Url', doc):
    print url.value
```

Buscando el libro con título = 'El color de la magia':

```
tree.find("Books/Book[Title='The Colour of Magic']")
# always use '' in the right side of the comparison
```