

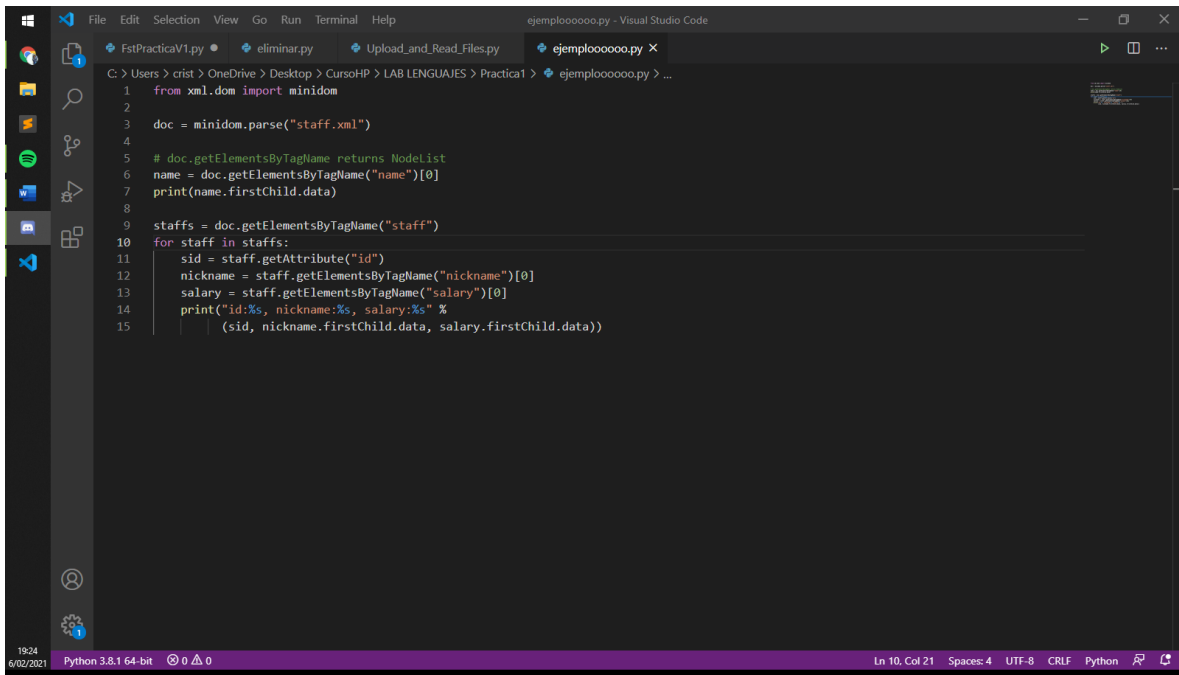
## DOM XML y XPath XML.

### DOM XML.

Esta es una recomendación del World Wide Web Consortium en la que todo el archivo se lee en la memoria y se almacena en una forma jerárquica (basada en árboles) para representar todas las características de un documento XML. Las aplicaciones DOM suelen comenzar analizando algún XML en un DOM.

El DOM es extremadamente útil para aplicaciones de acceso directo. SAX sólo te permite la vista de una parte del documento a la vez, debido a que algunas aplicaciones son imposibles en un modelo orientado a eventos sin acceso a un árbol, DOM es crea una representación del árbol estándar para datos XML.

**Funciones importantes (Parse() y parseString()):** Conectan un analizador sintáctico de XML con un «constructor DOM» que puede aceptar eventos de análisis de cualquier analizador sintáctico SAX y convertirlos en un árbol DOM, el análisis sintáctico del documento se completará antes de que retornen estas funciones, dichas funciones simplemente no proporcionan una implementación del analizador sintáctico por si mismas.



```
ejemploooooo.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
FstPracticaV1.py eliminar.py Upload_and_Read_Files.py ejemploooooo.py x
C:\Users\crist> OneDrive\ Desktop\ CursoHP\ LAB LENGUAJES\ Practica1\ ejemploooooo.py > ...
1 from xml.dom import minidom
2
3 doc = minidom.parse("staff.xml")
4
5 # doc.getElementsByTagName returns NodeList
6 name = doc.getElementsByTagName("name")[0]
7 print(name.firstChild.data)
8
9 staffs = doc.getElementsByTagName("staff")
10 for staff in staffs:
11     sid = staff.getAttribute("id")
12     nickname = staff.getElementsByTagName("nickname")[0]
13     salary = staff.getElementsByTagName("salary")[0]
14     print("id:%s, nickname:%s, salary:%s" %
15         (sid, nickname.firstChild.data, salary.firstChild.data))
```

19:24 6/02/2021 Python 3.8.1 64-bit Ln 10, Col 21 Spaces: 4 UTF-8 CRLF Python

```
File Edit Selection View Go Run Terminal Help • ejemploooooooo.py - Visual Studio Code
FstPracticaV1.py • eliminar.py • Upload_and_Read_Files.py • ejemploooooooo.py
C:\Users> crist > OneDrive > Desktop > CursoHP > LAB LENGUAJES > Practica1 > ejemploooooooo.py > ...
1 from xml.dom import minidom
2
3 doc = minidom.parse("staff.xml")
4
5 def getNodeText(node):
6
7     nodelist = node.childNodes
8     result = []
9     for node in nodelist:
10         if node.nodeType == node.TEXT_NODE:
11             result.append(node.data)
12     return ''.join(result)
13
14 name = doc.getElementsByTagName("name")[0]
15 print("Node Name : %s" % name.nodeName)
16 print("Node Value : %s \n" % getNodeText(name))
17
18
19 staffs = doc.getElementsByTagName("staff")
20 for staff in staffs:
21     sid = staff.getAttribute("id")
22     nickname = staff.getElementsByTagName("nickname")[0]
23     salary = staff.getElementsByTagName("salary")[0]
24     print("id:%s, nickname:%s, salary:%s" %
25         (sid, getNodeText(nickname), getNodeText(salary)))

19:30 6/02/2021 Python 3.8.1 64-bit 0 0 0 Ln 25, Col 65 Spaces: 4 UTF-8 CRLF Python
FstPracticaV1.py • eliminar.py • Upload_and_Read_Files.py • ejemploooooooo.py
C:\Users> crist > OneDrive > Desktop > CursoHP > LAB LENGUAJES > Practica1 > ejemploooooooo.py > ...
1 import xml.dom.minidom
2
3 document = """\
4 <slideshow>
5 <title>Demo slideshow</title>
6 <slide><title>Slide title</title>
7 <point>This is a demo</point>
8 <point>Of a program for processing slides</point>
9 </slide>
10
11 <slide><title>Another demo slide</title>
12 <point>It is important</point>
13 <point>To have more than</point>
14 <point>one slides</point>
15 </slide>
16 </slideshow>
17 """
18
19 dom = xml.dom.minidom.parseString(document)
20
21 def getText(nodelist):
22     rc = []
23     for node in nodelist:
24         if node.nodeType == node.TEXT_NODE:
25             rc.append(node.data)
26     return ''.join(rc)
27
28 def handleSlideshow(slideshow):
29     print("<html>")
30     handleSlideshowTitle(slideshow.getElementsByTagName("title")[0])
31     slides = slideshow.getElementsByTagName("slide")
32     handleToc(slides)
33     handleSlides(slides)
34     print("</html>")
```

## xPath XML.

Xpath es un módulo que es parte de la librería `xml.etree.ElementTree`, Xpath provee una serie de expresiones para localizar elementos en un árbol, su finalidad es proporcionar un conjunto de sintaxis, por lo que debido a su limitado alcance no se considera un motor en si mismo.

Permite extraer fácilmente partes de un XML haciendo referencia a su ubicación nodal representada a forma de path, simplificando el código.

### Xpath Syntaxis:

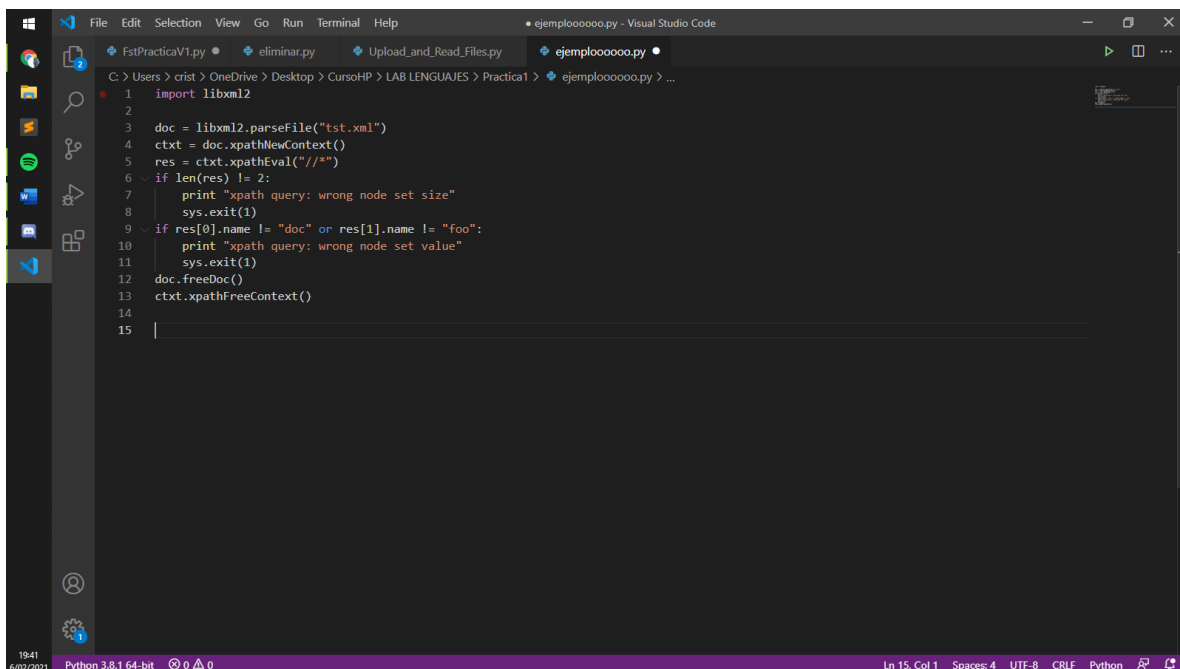
*Tag*: Selecciona todos los elementos hijos contenidos en la etiqueta "tag".

*\** : Selecciona todos los elementos hijos.

*[@attrib]*: Selecciona todos los elementos que contienen el atributo tras el "@" .

*[@attrib= "value"]*: Seleccione todos los elementos para los cuales el atributo dado tenga un valor dado, el valor no puede contener comillas.

*[tag]*: Selecciona todos los elementos que contienen una etiqueta hijo llamada tag. Solo los hijos inmediatos son admitidos



```
File Edit Selection View Go Run Terminal Help
ejemploooooo.py - Visual Studio Code
FstPracticaV1.py • eliminar.py Upload_and_Read_Files.py ejemploooooo.py •
C:\Users\crist > OneDrive > Desktop > CursoHP > LAB LENGUAJES > Practica1 > ejemploooooo.py > ...
1 import lxml2
2
3 doc = lxml2.parseFile("tst.xml")
4 ctxt = doc.xpathNewContext()
5 res = ctxt.xpathEval("/*")
6 if len(res) != 2:
7     print "xpath query: wrong node set size"
8     sys.exit(1)
9 if res[0].name != "doc" or res[1].name != "foo":
10    print "xpath query: wrong node set value"
11    sys.exit(1)
12 doc.freeDoc()
13 ctxt.xpathFreeContext()
14
15
```

1941 6/02/2021 Python 3.8.1 64-bit 0 0 0 Ln 15, Col 1 Spaces: 4 UTF-8 CRLF Python