

Unidade Curricular:**Integração de Sistemas de Informação****Tema da Ficha Prática:**

Utilização de XML, SCHEMAS, DTD, XPath, XQuery e XForms.

Objectivos:

Pretende-se com esta ficha prática que os alunos interajam com o conceito de XML e Schemas, XPath, XQuery e XForms.

Devem complementar esta informação com a disponibilizada pelos docentes nas aulas práticas.

Bibliografia:

Para apoio a esta ficha os alunos devem consultar os apontamentos teóricos e práticos da disciplina bem como de outros recursos online.

XML<https://www.w3.org/XML/>

The screenshot shows the official W3C XML page. At the top, there's a navigation bar with links for Home, XML Activity Statement, XML 1.0, XML 1.1, XML Schema, XML Namespaces, and XML 3.0. Below the navigation is a search bar and a link to Importar marcadores agora. The main content area has a blue header "Information and knowledge domain". A red arrow points from the text "This page describes the work being done at W3C within the XML Activity, and how it is structured." to the "Introduction" section. The "Introduction" section contains text about XML's history and its role in electronic publishing, followed by a list of links: 1. Introduction, 2. Working Groups, 3. Events, 4. Other Resources, and 5. Contact. To the right, there's a box titled "Upcoming:" with links to the XML London Conference and the Balisage Markup Conference. The URL in the address bar is https://www.w3.org/XML/.

FICHA PRÁTICA n.º 7

Integração de Sistemas de Informação

XML

http://ftp.sage.pt/sage/saft/Exemplo_SAFT_v1_01.xml

This XML file does not appear to have any style information associated with it. The document tree is as follows:

```
<AuditFile xmlns="urn:OECD:StandardAuditFile-Tax:PT_1_01_01">
  <Header>
    <DocumentID>SAFT_v1_01</DocumentID>
    <CompanyID>Conservas.pt</CompanyID>
    <TaxRegistrationNumber>123456789</TaxRegistrationNumber>
    <TaxAccountingBasis>I</TaxAccountingBasis>
    <CompanyName>Empresa de teste S.A.</CompanyName>
    <BusinessName>Empresa de testes</BusinessName>
  </Header>
  <CompanyAddress>
    <AddressDetail>Avenda das Demonstrações, lote 1 Piso 1</AddressDetail>
    <City>Póvoa das Demonstrações</City>
    <PostalCode>4100-615</PostalCode>
    <Country>PT</Country>
  </CompanyAddress>
  <FiscalYear>2008</FiscalYear>
  <StatementDate>2001-01-01</StatementDate>
  <ReportPeriod>2007-09-28</ReportPeriod>
  <CurrencyCode>EUR</CurrencyCode>
  <DateCreated>2007-09-28</DateCreated>
  <TaxEntitySede>TaxEntity</TaxEntity>
  <ProductCompanyTaxID>123456789</ProductCompanyTaxID>
  <SoftwareCertificateNumber>0</SoftwareCertificateNumber>
  <ProductVersion>saft 0.1.0004</ProductVersion>
  <HeaderComment>Comentários ao SAFT exportado</HeaderComment>
  <Telephone>211234567</Telephone>
  <Fax>211234567</Fax>
  <Email>testemaria.com</Email>
  <Website>www.mariatestes.com</Website>
  </Header>
  <HeaderLines>
    <GeneralLedger>
      <AccountID>22</AccountID>
      <AccountDescription>Fornecedores</AccountDescription>
      <OpeningDebitBalance>0</OpeningDebitBalance>
      <OpeningCreditBalance>0</OpeningCreditBalance>
    </GeneralLedger>
    <AccountID>221</AccountID>
    <AccountDescription>Fornecedores c/c</AccountDescription>
    <OpeningDebitBalance>0</OpeningDebitBalance>
    <OpeningCreditBalance>0</OpeningCreditBalance>
  </HeaderLines>
  <GeneralLedger>
    <AccountID>24</AccountID>
    <AccountDescription>Estado e Outros Entes Públicos</AccountDescription>
    <OpeningDebitBalance>0</OpeningDebitBalance>
    <OpeningCreditBalance>0</OpeningCreditBalance>
  </GeneralLedger>

```



Não seguro | http://ftp.sage.pt/sage/saft/Exemplo_SAFT_v1_01.xml

```
<AuditFile xmlns="urn:OECD:StandardAuditFile-Tax:PT_1_01_01">
  <Header>
    <DocumentID>SAFT_v1_01</DocumentID>
    <CreditAmount>130</CreditAmount>
  </Header>
  <Tax>
    <TaxType>IVA</TaxType>
    <TaxCountryRegion>PT</TaxCountryRegion>
    <TaxCode>NR</TaxCode>
    <TaxPercentage>21</TaxPercentage>
  </Tax>
  <SettlementAmount>0</SettlementAmount>
  </Line>
  <DocumentTotals>
    <TaxPayable>128.44</TaxPayable>
    <NetTotal>612.59</NetTotal>
    <GrossTotal>741.23</GrossTotal>
  </DocumentTotals>
  <Invoice>
    <InvoiceNo>NF 1/2</InvoiceNo>
    <InvoiceStatus>N</InvoiceStatus>
    <InvoiceID>123456789</InvoiceID>
    <Period>9</Period>
    <InvoiceDate>2007-09-28</InvoiceDate>
    <InvoiceType>FT</InvoiceType>
    <SalesBillingIndicator>1</SalesBillingIndicator>
    <SystemEntryDate>2007-09-28T15:31:21</SystemEntryDate>
    <CustomerID>CL</CustomerID>
    <Line>
      <LineNumber>1</LineNumber>
      <ProductCode>P2</ProductCode>
      <ProductDescription>Canetas</ProductDescription>
      <Quantity>2</Quantity>
      <UnitOfMeasure>Unid</UnitOfMeasure>
      <UnitPrice>1.39</UnitPrice>
      <TaxPointDate>2007-09-28</TaxPointDate>
      <Description>Canetas</Description>
      <CreditAmount>2.7</CreditAmount>
    </Line>
    <Tax>
      <TaxType>IS</TaxType>
      <TaxCountryRegion>PT</TaxCountryRegion>
      <TaxCode>2.1.1</TaxCode>
      <TaxPercentage>21</TaxPercentage>
    </Tax>
  </Invoice>
  <SalesInvoices>
    <SourceDocuments>
  </SalesInvoices>
  </SourceDocuments>
</AuditFile>
```

<https://www.w3.org/standards/xml/schema.html>

w3.org/standards/xml/schema.html

Leading the web to its full potential

STANDARDS PARTICIPATE MEMBERSHIP ABOUT W3C

W3C » Standards » Web Design and Applications » XML Technology » Schema

SCHEMA

On this page → what is an XML Schema what is an XML Schema used for examples

An XML Schema is a language for expressing constraints about XML documents.

What is an XML Schema?

An XML Schema is a language for expressing constraints about XML documents. There are several different schema languages in widespread use, but the main ones are Document Type Definitions (DTDs), Relax-NG, Schematron and W3C XSD (XML Schema Definitions). From this page you can find out more about DTDs and W3C XSD, since those are the primary schema languages defined at W3C.

What is XML Schema Used For?

A Schema can be used:

What is XML Schema Used For?

A Schema can be used:

- to provide a list of elements and attributes in a vocabulary;
- to associate types, such as integer, string, etc., or more specifically such as hatsize, sock_colour, etc., with values found in documents;
- to constrain where elements and attributes can appear, and what can appear inside those elements, such as saying that a chapter title occurs inside a chapter, and that a chapter must consist of a chapter title followed by one or more paragraphs of text;
- to provide documentation that is both human-readable and machine-processable;
- to give a formal description of one or more documents.

Information in schema documents is often used by XML-aware editing systems so that they can offer users the most likely elements to occur at any given location in a document.

Checking a document against a Schema is known as validating against that schema; for a DTD, this is just *validating*, but for any other type of schema the type is mentioned, such as XSD Validation or Relax-NG validation.

Validating against a schema is an important component of quality assurance.

The screenshot shows a web browser window with the URL whatis.techtarget.com/definition/XSD-XML-Schema-Definition. The page has a teal header with the TechTarget logo and the WhatIs.com logo. It features a navigation bar with links for 'BROWSE DEFINITIONS' (AppDev), 'QUICK STUDY Resources', and a search bar. Below the header, there's a 'DEFINITION' section for 'XSD (XML Schema Definition)'. The definition is written by George Lawton and includes social sharing icons for Facebook, Twitter, LinkedIn, and Print. The text explains that XSD is a W3C recommendation for formally describing XML documents, and it notes that XSD 1.1 became an approved standard in April 2012. A sidebar on the left contains icons for various definitions and resources.

XSD (XML Schema Definition)

By [George Lawton](#)

XSD (XML Schema Definition) is a World Wide Web Consortium ([W3C](#)) recommendation that specifies how to formally describe the elements in an Extensible Markup Language (XML) document. This description can be used to verify that each item of content in a document adheres to the description of the element in which the content is to be placed. XSD 1.1 became an approved W3C standard in April 2012.

XSD can also be used for generating XML documents that can be treated as programming objects. In addition, a variety of XML processing tools can also generate human readable documentation, which makes it easier to understand complex XML documents.

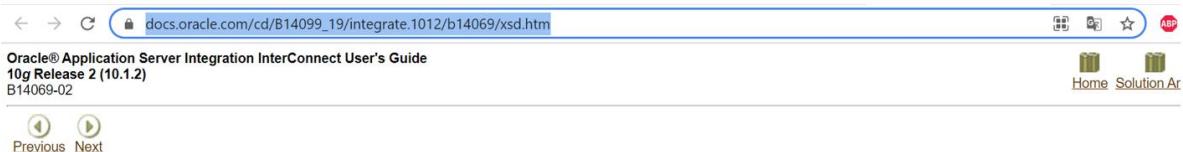
XSD Example

```
<?xml version="1.0"?>
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema">

<xss:element name="note">
  <xss:complexType>
    <xss:sequence>
      <xss:element name="to" type="xss:string"/>
      <xss:element name="from" type="xss:string"/>
      <xss:element name="heading" type="xss:string"/>
      <xss:element name="body" type="xss:string"/>
    </xss:sequence>
  </xss:complexType>
</xss:element>

</xss:schema>
```

https://docs.oracle.com/cd/B14099_19/integrate.1012/b14069/xsd.htm



The screenshot shows a web browser displaying the Oracle Application Server Integration InterConnect User's Guide. The URL in the address bar is docs.oracle.com/cd/B14099_19/integrate.1012/b14069/xsd.htm. The page content is identical to the one shown above, containing the XML schema code for a 'note' element.

10 XML Schema Definition

This chapter provides an overview of XML Schema Definition (XSD) and describes how InterConnect supports XSD. It contains the following topics:

- [Overview of XSD](#)
- [XSD Type Definitions](#)
- [XSD Elements and Attributes](#)
- [XSD Content Models](#)
- [XSD Namespace](#)
- [XSD Type Derivations](#)
- [Unsupported Features and Limitations of XSD](#)

Overview of XSD

An XSD defines the structure of an XML document. It specifies the elements and attributes that can appear in an XML document and the type of data these elements and attributes can contain. This to verify that each element or attribute in an XML document adheres to its description.

Advantages of XSD over DTD

Following are the advantages of XSD over Document Type Definition (DTD):

- XSD is extensible. You can derive new elements from the existing elements. DTD is not extensible.
- XSD is defined in XML. It does not require intermediate processing by a parser. DTD is not defined in XML. You need separate parsers for DTD and XML.
- XSD supports data types. You can restrict the content of an element. DTD does not support data types. Therefore, you cannot restrict the content of an element.

FICHA PRÁTICA n.º 7

Integração de Sistemas de Informação

<https://docs.microsoft.com/en-us/dotnet/standard/data/xml/working-with-xml-schemas>

The screenshot shows a Microsoft Docs page titled "Working with XML Schemas". The sidebar contains a tree view of topics under "WORKING WITH XML SCHEMAS", including "Working with XML Schemas", "XML Schema Object Model (SOM)", "XmlSchemaSet for Schema Compilation", "XmlSchemaValidator Push-Based Validation", "Inferring an XML Schema", "XML Integration with Relational Data and ADO.NET", and "Managing Namespaces in an XML". Below the sidebar is a "Download PDF" button. The main content area starts with a brief introduction about XML Schema definition language (XSD) and its role in defining XML document structure. It mentions that while well-formed XML documents are valid, not all valid XML documents are well-formed. It also links to the W3C XML 1.0 Recommendation and the W3C XML Schema Part 1: Structures Recommendation. The "Resulting XML sample" section displays the following XML code:

```
<?xml version="1.0" encoding="utf-8"?>
<note>
  <to>str1234</to>
  <from>str1234</from>
  <heading>str1234</heading>
  <body>str1234</body>
</note>
```

<https://docs.microsoft.com/en-us/visualstudio/xml-tools/how-to-create-an-xml-document-based-on-an-xsd-schema?view=vs-2019>

The Generate Sample XML feature generates a sample XML file based on your XML Schema (XSD) file. You can use this option for the following scenarios:

- To understand the use of various constructs in your schema.
- To confirm that the schema does what it is intended to do.

The Generate Sample XML feature is only available on global elements, and requires a valid XML schema set. This feature typically generates valid XML documents. However, if the schema contains one or more of the following, the sample might not be valid:

- The `xs:key`, `xs:keyref`, and `xs:unique` identity constraints.
- `xs:pattern` facets.
- Enumerations of the `xs:QName` type.



XPointer

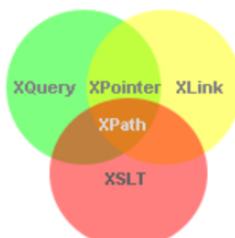


- XPointer allows links to point to specific parts of an XML document
- XPointer uses XPath expressions to navigate in the XML document
- XPointer is a W3C Recommendation

https://www.w3schools.com/xml/xml_xlink.asp

<https://www.w3.org/TR/WD-xptr>

XLink is used to create hyperlinks in XML documents.



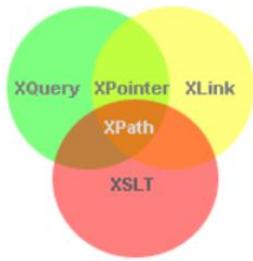
- XLink is used to create hyperlinks within XML documents
- Any element in an XML document can behave as a link
- With XLink, the links can be defined outside the linked files
- XLink is a W3C Recommendation

<https://www.w3.org/TR/xlink/>

https://www.w3schools.com/xml/xml_xlink.asp

FICHA PRÁTICA n.º 7

Integração de Sistemas de Informação



- XPath stands for XML Path Language
- XPath uses "path like" syntax to identify and navigate nodes in an XML document
- XPath contains over 200 built-in functions
- XPath is a major element in the XSLT standard
- XPath is a W3C recommendation

XPATH – TUTORIAIS:

<https://librarycarpentry.org/lc-webscraping/02-xpath/index.html>

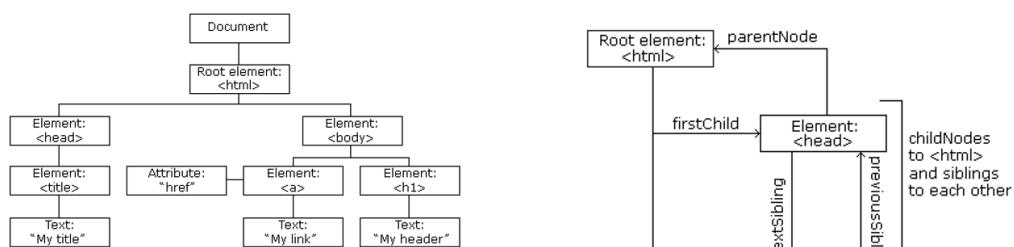
librarycarpentry.org/lc-webscraping/02-xpath/index.html

XPath always assumes *structured* data.

Now let's start using XPath.

Navigating through the HTML node tree using XPath

A popular way to represent the structure of an XML or HTML document is the *node tree*:



In an HTML document, everything is a node:

- The entire document is a document node
- Every HTML element is an element node
- The text inside HTML elements are text nodes

The nodes in such a tree have a hierarchical relationship to each other. We use the terms *parent*, *child* and *sibling* to describe these relationships:

- In a node tree, the top node is called the *root* (or *root node*)
- Every node has exactly one *parent*, except the root (which has no parent)
- A node can have zero, one or several *children*
- *Siblings* are nodes with the same parent
- The sequence of connections from node to node is called a *path*

XPATH

<https://www.w3.org/TR/xpath/>

This screenshot shows the W3C XPATH COVER PAGE. The page header includes the W3C logo and navigation links for STANDARDS, PARTICIPATE, MEMBERSHIP, and ABOUT W3C. The main content area displays a table of XPath standards:

XPATH COVER PAGE			
2017-03-21	Recommendation	XML Path Language (XPath) 3.1 XPath is an expression language that allows the processing of values conforming to the data model defined in the XQuery and XPath Model.	
2014-04-08	Recommendation	XML Path Language (XPath) 3.0 XPath 3.0 (renamed from XPath 2.1 to align with the family of "3.0" specifications) is an expression language that allows the process values conforming to the data model defined in [XQuery and XPath Data Model (XDM) 3.0]. Some of the important new features since XPath 2.0 are: Literal function items, inline functions, dynamic function invocations, and function item coercion Clarification of rules associated with sequence type matching let expressions QNames (QNames with a namespace URI instead of a namespace prefix) Support for union types in casting and function arguments	
2010-12-14	Recommendation	XML Path Language (XPath) 2.0 (Second Edition) XPath is a way to refer to parts of an XML document. XPath 2.0 is based on the XQuery 1.0 and XPath 2.0 Data Model (XDM), and it introduces Schema awareness and data typing.	
1999-11-16	Recommendation	XML Path Language (XPath) Version 1.0 XPath is a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer.	

XPATH – TUTORIAIS:

<https://docs.scrapy.org/en/xpath-tutorial/topics/xpath-tutorial.html>

HTML Input

```
<html>
<head>
<title>This is a title</title>
<meta content="text/html; charset=utf-8" http-equiv="content-type" />
</head>
<body>
<div>
<p>This is a paragraph.</p>
<p>Is this <a href="page2.html">a link</a>?</p>
<br />.
.
```

XPath Expression

```
//body//text()
```

Result

```
This is a paragraph.
Is this
a link
```

HTML Input

```
<html>
<head>
<title>This is a title</title>
<meta content="text/html; charset=utf-8" http-equiv="content-type" />
</head>
<body>
<div>
<p>This is a paragraph.</p>
<p>Is this <a href="page2.html">a link</a>?</p>
<br />.
.
```

XPath Expression

```
/a/@href
```

Result

```
page2.html
page3.html
```

XPATH – TUTORIAIS:

- O que é o XPATH: <https://www.javatpoint.com/xpath-interview-questions>
- Tutorial 1: http://www.macoratti.net/vb_xpath.htm
- Tutoria 2: https://www.tutorialspoint.com/xpath/xpath_expression.htm
- Tutorial 3: <https://www.softwaretestinghelp.com/xml-path-language-xpath-tutorial/>
- Intro to XPath with Java: <https://www.baeldung.com/java-xpath>
- Java XPath Parser - Parse XML Document:https://www.tutorialspoint.com/java_xml/java_xpath_parse_document.htm
- EXECUTAR O TUTORIAL: Java XPath Example – XPath Tutorial:
<https://howtodoinjava.com/java/xml/java-xpath-tutorial-example/>

Parsing XML

We'll be using the following XML document as the sample data for this section:

```
<?xml version="1.0"?>
<data>
    <country name="Liechtenstein">
        <rank>1</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

We can import this data by reading from a file:

```
import xml.etree.ElementTree as ET
tree = ET.parse('country_data.xml')
root = tree.getroot()
```

Or directly from a string:

```
root = ET.fromstring(country_data_as_string)
```

`fromstring()` parses XML from a string directly into an `Element`, which is the root element of the parsed tree. Other parsing functions may create an `ElementTree`. Check the documentation to be sure.

As an `Element`, `root` has a tag and a dictionary of attributes:

```
>>> root.tag
'data'
>>> root.attrib
{}
```

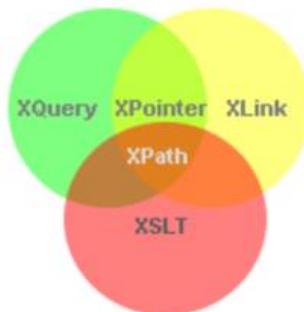
It also has children nodes over which we can iterate:

```
>>> for child in root:
...     print(child.tag, child.attrib)
...
country {'name': 'Liechtenstein'}
country {'name': 'Singapore'}
country {'name': 'Panama'}
```

Children are nested, and we can access specific child nodes by index:

```
>>> root[0][1].text
'2008'
```

What is XQuery?



- XQuery is **the** language for querying XML data
- XQuery for XML is like SQL for databases
- XQuery is built on XPath expressions
- XQuery is supported by all major databases
- XQuery is a W3C Recommendation

XQUERY

<https://www.w3.org/XML/Query/>

The un-queried life is not worth living.
— Socrates (*Plato, The Apology*, 3)

High level Overview Architects & Analysts For Users For Implementers What's New References

Tutorial:

- <http://www.cs.toronto.edu/~leijiang/teaching/cscc43-s09/content/tutorial/xpath-xquery/xquery.pdf>
- https://www.tutorialspoint.com/xquery/xquery_xpath.htm
- https://docs.safe.com/fme/html/FME/Desktop_Documentation/FME_Transformers/xquery_samples.htm

w3.org/XML/Query/#implementations

IMPLEMENTATIONS

XML Query Implementations

If your implementation is not here, or if you know of an implementation that is not listed, send liam@w3.org the details!

Software that implements the XQuery and XPath Full Text Facility is so marked in this list. There is a separate list of [XPath 2](#) implementations.

1. The W3C [XPath and XQuery Grammar Test Page](#) features Java applets that read expressions and show the resulting parse tree.
2. Abacus Systems' [Relational XQuery](#) supports both relational data (via JDBC) and other sources including XML files, and also claims XQJ (XQuery for Java API) conformance. Includes a GUI for creating and editing queries. 30 day evaluation. [2011-03-25: Last update July 2009; project homepage is gone.]
3. Altova GmbH [XMLSpy 2006](#) includes an XQuery Debugger, a code generator for mapping between Schemas, and [AltovaXML Query Processor](#) which handles both XSLT 2 and XML Query 1.0 [30-day free trial]
4. Apple's [Sherlock](#) for Mac OS X; see also their [XML Query Extension functions](#).
5. BEA's [Oracle Data Services Platform](#) [90-day free trial]. BEA was an active participant in the XML Query Working Group; the company was bought by Oracle, also an active participant.
6. Berkeley Lab's [Nux](#), an open source Java in-memory toolkit for XML, XQuery, XPath, schema validation, fuzzy fulltext similarity search and related technologies using Saxon, XOM, Xerces and JAXB [open source under a BSD-style license]. Implements the XQuery Update Facility; Full-Text Support; latest release seems to be June 2006.
7. Bluestream Database Software Corp.'s [XStreamDB](#), a native XML database server and full text support, aimed primarily at DITA. [commercial with trial download]
8. David Carlisle's [xq2xml](#) converts XQuery to XML, to XQueryX and to XSLT.
9. Cerebra Inc.'s Cerebra Server supports XQuery, OWL-DL and RDF, and can connect to external databases, but their Web server no longer responds.
10. Cognitive Systems's [XQuantum](#) implements XML Query 1.0 in an XML-native indexed data store. They have a Web page demonstrating the XQuery Use Cases, and support static typing and modules as well as some full-text extensions. [Windows and Linux; 30-day evaluation] Full-Text Support
11. DataDirect's [DataDirect XQuery \(tm\)](#), an embeddable component for XQuery that implements the XQuery for Java(tm) API (XQJ) [Java; 15-day trial download].

<https://www.javatpoint.com/difference-between-xquery-and-xpath>

difference-between-xquery-and-xpath

HTML CSS JavaScript jQuery XML XSLT XPath XQuery JSON Ajax Quiz Project

XQuery vs XPath

← Prev Next →

Index	XQuery	XPath
1)	XQuery is a functional programming and query language that is used to query a group of XML data.	XPath is a XML path language that is used to select nodes from an XML document using queries.
2)	XQuery is used to extract and manipulate data from either XML documents or relational databases and MS Office documents that support an XML data source.	XPath is used to compute values like strings, numbers and boolean types from another XML documents.
3)	xquery is represented in the form of a tree model with seven nodes, namely processing instructions, elements, document nodes, attributes, namespaces, text nodes, and comments.	xpath is represented as tree structure, navigate it by selecting different nodes.

4)	xquery supports xpath and extended relational models.	xpath is still a component of query language.
5)	xquery language helps to create syntax for new XML documents.	xpath was created to define a common syntax and behavior model for xpointer and xslt.

FICHA PRÁTICA n.º 7

Integração de Sistemas de Informação

books.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<books>

<book category="JAVA">
    <title lang="en">Learn Java in 24 Hours</title>
    <author>Robert</author>
    <year>2005</year>
    <price>30.00</price>
</book>

<book category="DOTNET">
    <title lang="en">Learn .Net in 24 hours</title>
    <author>Peter</author>
    <year>2011</year>
    <price>40.50</price>
</book>

<book category="XML">
    <title lang="en">Learn XQuery in 24 hours</title>
    <author>Robert</author>
    <author>Peter</author>
    <year>2013</year>
    <price>50.00</price>
</book>

<book category="XML">
    <title lang="en">Learn XPath in 24 hours</title>
    <author>Jay Ban</author>
    <year>2010</year>
    <price>16.50</price>
</book>

</books>
```

XQuery Expression (Index)

```
let $items := (1,2,3,4,5,6)
let $count := count($items)
return
<result>
    <count>{$count}</count>

    <items>
    {
        for $item in $items[2]
        return <item>{$item}</item>
    }
    </items>

</result>
```

Output

```
<result>
    <count>6</count>
    <items>
        <item>2</item>
    </items>
</result>
```

XQuery – Version 1

```
(: read the entire xml document :)
let $books := doc("books.xml")

for $x in $books/books/book
where $x/price > 30
return $x/title
```

Output

```
<title lang="en">Learn .Net in 24 hours</title>
<title lang="en">Learn XQuery in 24 hours</title>
```

XQuery – Version 2

```
(: read all books :)
let $books := doc("books.xml")/books/book

for $x in $books
where $x/price > 30
return $x/title
```

Output

```
<title lang="en">Learn .Net in 24 hours</title>
<title lang="en">Learn XQuery in 24 hours</title>
```

XQuery Expression (Value)

```
let $items := (1,2,3,4,5,6)
let $count := count($items)
return
<result>
    <count>{$count}</count>

    <items>
    {
        for $item in $items[. = (1,2,3)]
        return <item>{$item}</item>
    }
    </items>

</result>
```

Output

```
<result>
    <count>6</count>
    <items>
        <item>1</item>
        <item>2</item>
        <item>3</item>
    </items>
</result>
```

FICHA PRÁTICA n.º 7**Integração de Sistemas de Informação**

[←](#) [→](#) [C](#) [javatpoint.com/xquery-first-example](#) [QR](#) [🔍](#) [☆](#) [4](#)

Home C Java PHP HTML CSS JavaScript jQuery XML XSLT XPath XQuery JSON Ajax Quiz Projects Interview Q Comment Forum Training

[Environment Setup](#) [XQuery First Example](#) [XQuery FLWOR](#) [XQuery HTML Format](#) [XQuery XPath](#) [XQuery Syntax](#) [XQuery Add](#) [XQuery Functions](#) [XQuery Sequences](#) [XQuery Sequence functions](#) [XQuery String functions](#) [1\) String-length\(\)](#) [2\) String concat\(\)](#) [3\) String-join\(\)](#) [XQuery Time & Date](#) [1\) current-date\(\)](#) [2\) current-time\(\)](#) [3\) current-dateTime\(\)](#) [XQuery If Then Else](#) [XQuery Regex](#)

XQuery First Example [← Prev](#) [Next →](#)

Here, the XML document is named as **courses.xml** and xqy file is named as **courses.xqy**

courses.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<courses>
  <course category="JAVA">
    <title lang="en">Learn Java in 3 Months.</title>
    <trainer>Sonoo Jaiswal</trainer>
    <year>2008</year>
    <fees>10000.00</fees>
  </course>
  <course category="Dot Net">
    <title lang="en">Learn Dot Net in 3 Months.</title>
    <trainer>Vicky Kaushal</trainer>
    <year>2008</year>
    <fees>10000.00</fees>
  </course>
  <course category="C">
    <title lang="en">Learn C in 3 Months</title>
    <trainer>John Smith</trainer>
    <year>2008</year>
    <fees>10000.00</fees>
  </course>
</courses>
```

[SCROLL TO TOP](#)

courses.xqy

```
for $x in doc("courses.xml")/courses/course
where $x/fees>5000
return $x/title
```

This example will display the title elements of the courses whose fees are greater than 5000.

Create a Java based XQuery executor program to read the courses.xqy, passes it to the XQuery expression processor, and executes the expression. After that the result will be displayed.

XQueryTester.java

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;

import javax.xml.xquery.XQConnection;
import javax.xml.xquery.XQDataSource;
```

XQueryTester.java

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;

import javax.xml.xpath.XQConnection;
import javax.xml.xpath.XQDataSource;
import javax.xml.xpath.XQException;
import javax.xml.xpath.XQPreparedExpression;
import javax.xml.xpath.XQResultSequence;

import com.saxonica.xqj.SaxonXQDataSource;

public class XQueryTester {
    public static void main(String[] args){
        try {
            execute();
        }
        catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

```
    catch (XQException e) {
        e.printStackTrace();
    }
}

private static void execute() throws FileNotFoundException, XQException{
    InputStream inputStream = new FileInputStream(new File("courses.xqy"));
    XQDataSource ds = new SaxonXQDataSource();
    XQConnection conn = ds.getConnection();
    XQPreparedExpression exp = conn.prepareExpression(inputStream);
    XQResultSequence result = exp.executeQuery();
    while (result.next()) {
        System.out.println(result.getItemAsString(null));
    }
}
}
```

Execute XQuery against XML

Put the above three files to a same location. We put them on desktop in a folder name XQuery2. Compile XQueryTester.java using console. You must have JDK 1.5 or later installed on your computer and classpaths are configured.

Compile:

```
javac XQueryTester.java
```

Execute:

```
java XQueryTester
```

Output:

```
C:\Windows\system32\cmd.exe
C:\Users\javatpoint1\Desktop>cd XQuery2
C:\Users\javatpoint1\Desktop\XQuery2>javac XQueryTester.java
C:\Users\javatpoint1\Desktop\XQuery2>java XQueryTester
<title lang="en">Learn Java in 3 Months.</title>
<title lang="en">Learn Dot Net in 3 Months.</title>
C:\Users\javatpoint1\Desktop\XQuery2>
```

FICHA PRÁTICA n.º 7**Integração de Sistemas de Informação**

<https://docs.python.org/3/library/xml.etree.elementtree.html>

The `xml.etree.ElementTree` module implements a simple and efficient API for parsing and creating XML data.

Changed in version 3.3: This module will use a fast implementation whenever available.

Deprecated since version 3.3: The `xml.etree.cElementTree` module is deprecated.

Warning: The `xml.etree.ElementTree` module is not secure against maliciously constructed data. If you need to parse untrusted or unauthenticated data see [XML vulnerabilities](#).

<https://doc.qt.io/qtforpython-5/overviews/xmlprocessing.html>

An overview of Qt's support for using XML technologies in Qt programs.

Introduction

`XQuery` is a language for traversing XML documents to select and aggregate items of interest and to transform them for output as XML or some other format. XPath is the *element selection* part of `XQuery`.

The Qt XML Patterns module supports using `XQuery 1.0` and `XPath 2.0` in Qt applications, for querying XML data *and for* querying non-XML data that can be modeled to look like XML. Readers who are not familiar with the `XQuery` /XPath language can read [A Short Path to XQuery](#) for a brief introduction.

Advantages of Using Qt XML Patterns and XQuery

The `XQuery` /XPath language simplifies data searching and transformation tasks by eliminating the need for doing a lot of C++ or Java procedural programming for each new query task. Here is an `XQuery` that constructs a bibliography of the contents of a library:

FICHA PRÁTICA n.º 7

Integração de Sistemas de Informação

<https://python.hotexamples.com/examples/xutil.xquery/XQuery/-/python-xquery-class-examples.html>

The screenshot shows a web browser with the URL https://python.hotexamples.com/examples/xutil.xquery/XQuery/-/python-xquery-class-examples.html. The page title is "Python XQuery Examples". Below the title, it says "Python XQuery - 21 examples found. These are the top rated real world Python examples of `xutilxquery.XQuery` extracted from open source projects. You can rate help us improve the quality of examples." It lists the programming language as Python, namespace/package name as `xutilxquery`, and class/type as XQuery. It also mentions "Examples at hotexamples.com: 21". On the left, there's a sidebar with "FREQUENTLY USED METHODS" like `find`, `remove`, `append`, etc., and a "RELATED" section with links like `CharacterAwardComponent`, `cache_reg_apps`, etc. The main content area shows "EXAMPLE #1" with code snippets and line numbers.

Python XQuery Examples

Python XQuery - 21 examples found. These are the top rated real world Python examples of `xutilxquery.XQuery` extracted from open source projects. You can rate help us improve the quality of examples.

Programming Language: Python

Namespace/Package Name: `xutilxquery`

Class/Type: XQuery

Examples at hotexamples.com: 21

FREQUENTLY USED METHODS

- `find` (14)
- `remove` (4)
- `append` (1)
- `clone` (1)
- `is_etree_node_eq` (1)

RELATED

- `CharacterAwardComponent`
- `cache_reg_apps`
- `in_main_thread`
- `urllib_directory`

EXAMPLE #1

Show file

File: `test_xquery.py` Project: huanghao/xutil

```
def test_remove_desendent(self):
    x = XQuery('''<R><a>1</a><b><c>3</c></b></R>''')

    self.assertEqual(XQuery('<c>3</c>'),
                    x.remove('c'))
    self.assertEqual(XQuery('<R><a>1</a><b></b></R>'),
                    x)
```

0

<https://www.code-learner.com/python-parse-html-page-with-xpath-example/>

The screenshot shows a web browser with the URL https://www.code-learner.com/python-parse-html-page-with-xpath-example/. The page title is "Python Parse Html Page With XPath Example". Below the title, it says "Leave a Comment / Python". The main content area contains a text block explaining how Python can be used to parse HTML pages using the `lxml` library's `xpath` method. It also includes a snippet of Python code for parsing an HTML page.

Home » Python » Python Parse Html Page With XPath Example

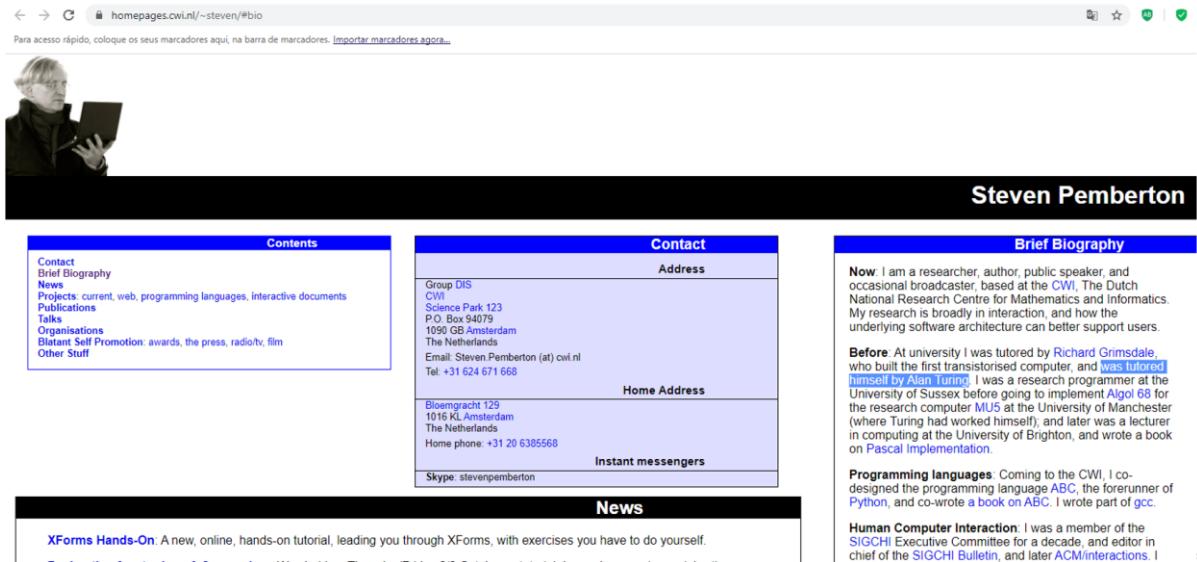
Python Parse Html Page With XPath Example

[Leave a Comment](#) / Python

Python can be used to write a web page crawler to download web pages. But the web page content is massive and not clear for us to use, we need to filter out the useful data that we need. This article will tell you how to parse the downloaded web page content and filter out the information you need use the python `lxml` library's `xpath` method.

When it comes to string content filtering, we immediately think about regular expressions, but we won't talk about regular expressions today. Because regular expressions are too complex for a crawler that is written by a novice. Moreover, the error tolerance of regular expressions is poor,



FICHA PRÁTICA n.º 7**Integração de Sistemas de Informação****XFORMS: Prof. Steven Pemberton**<https://homepages.cwi.nl/~steven/#bio>


The screenshot shows a web browser displaying a personal homepage. At the top, there is a navigation bar with icons for back, forward, search, and other functions. Below it, a message says "Para acesso rápido, coloque os seus marcadores aqui, na barra de marcadores. Importar marcadores agora...". The main content area features a large portrait of Steven Pemberton holding a book. To the right of the portrait, the name "Steven Pemberton" is displayed in a bold black font. Below the portrait, there are three columns of information:

- Contents** (left column):
 - Contact
 - Brief Biography
 - News
 - Projects: current, web, programming languages, interactive documents
 - Publications
 - Talks
 - Organisations
 - Blatant Self Promotion: awards, the press, radio/tv, film
 - Other Stuff
- Contact** (middle column):

Address
Group DIS CWI Science Park 123 P.O. Box 94079 1090 GK Amsterdam The Netherlands Email: Steven.Pemberton (at) cwi.nl Tel. +31 624 671 668
Home Address
Bloemgracht 129 1016 KL Amsterdam The Netherlands Home phone: +31 20 6305568
Instant messengers
Skype: stevenpemberton
- Brief Biography** (right column):

Now, I am a researcher, author, public speaker, and occasional broadcaster, based at the [CWI](#). The Dutch National Research Centre for Mathematics and Informatics. My research is broadly in interaction, and how the underlying software architecture can better support users.

Before: At university I was tutored by [Richard Grimsdale](#), who built the first transistorised computer, and [was tutored himself by Alan Turing](#). I was a research programmer at the University of Sussex before going to implement [Algol 68](#) for the research computer [MUS](#) at the University of Manchester (where Turing had worked himself), and later was a lecturer in computing at the University of Brighton, and wrote a book on [Pascal implementation](#).

Programming languages: Coming to the CWI, I co-designed the programming language [ABC](#), the forerunner of [Python](#), and co-wrote a book on [ABC](#). I wrote part of [gcc](#).

Human Computer Interaction: I was a member of the [SIGCHI Executive Committee](#) for a decade, and editor in chief of the [SIGCHI Bulletin](#), and later [ACM/Interactions](#). I

At the bottom left, there is a "News" section with a link to "XForms Hands-On: A new, online, hands-on tutorial, leading you through XForms, with exercises you have to do yourself." At the bottom right, there are "Prev" and "Next" navigation buttons.

XForms Features[← Prev](#)[Next →](#)

- XForms is a new generation of HTML forms. It inherits some properties of HTML forms and extends some other properties. In other word, XForms can be called the successor of HTML forms.
- XForms uses XML for data definition and HTML or XHTML to display data.
- XForms separates the data logic of the form from its presentation.
- XForms uses XML to define form data.
- XForms uses XML to store and transport data. The data displayed in a form are stored in an XML document, and the data submitted from the form, are transported over the internet using XML.
- XForms is device independent because data is separated from presentation and the data model can be used for all devices.
- XForms is a W3C recommendation.

XFORMS: Prof. Steven Pemberton

<https://homepages.cwi.nl/~steven/#bio>

← → ⌂ homepages.cwi.nl/~steven/#bio

Para acesso rápido, coloque os seus marcadores aqui, na barra de marcadores. [Importar marcadores agora...](#)

Guilty: Great work!

"Love the elaboration of numbers as a concept. ❤️" "So pleasant to read, very accessible!" "It's great!"

Onze maatschappij is nog niet klaar voor AI: Interview with me (in Dutch) in AGConnect. (Alas registration required to read it all).

"A back-end engineer gets overly excited about stack machines..." and other talks...: an interview with me and two other speakers at Bristech.

Inspiring Web Pioneer Steven Pemberton: an interview.

A series of articles on XForms I am writing, published at XML.com.

"Even as someone who's not a developer, I've found [these] quite approachable."

- **An Introduction to XForms:** An overview and a simple example. ["Fabulous"](#)
- **Viewing Data with XForms:** Displaying and searching data.
- **A Calendar in XForms:** Dynamic data. ["Cool!"](#) ["Very nice!"](#) ["Really very neat!"](#) ["XForms awesomeness!"](#)
- **A Clock in XForms:** XForms plus SVG. ["Great tutorial, and very cool example!"](#)
- **A Game in XForms:** Interaction.
- **A News Carousel in XForms:** Displaying alternating information.
- **NoPHP: A Conference Website in XForms:** Managing and updating live information. ["Another great step-by-step demo showing how easy it is to do things in XForms."](#)

Want to learn XForms? See [XForms: An Introduction](#).

Lit Lace: A colleague and I have worked on a project to design and implement interfaces to self-illuminating fabrics. It is [featured in Dutch Design Week](#) this year.

Invisible XML Specification: a draft is available. Comments gladly received.

Upcoming and recent talks:

Invited Talk: Declarative Programming at 79th meeting of IFIP WG 2.1, Otterlo, NL, 6-10 January 2020.

Invited Talk: The Internet of Things and the Coming Robot Rebellion at Freelance Friday, Pakhuis de Zwijger Amsterdam, NL, 10 January 2020. ["Really great talk!"](#) ["even intrigerend als verontrustend!"](#) ["Inspirational and mind twisting!"](#) ["super interessant verhaal dat je dwingt tot kijken vanuit verschillende perspectieven!"](#)

Tutorial: Declarative Applications with XForms at XML Prague, Prague, Czechia, 14-15 February. ["really interesting!"](#) ["It is not often that someone who is technically proficient in his field is also a good speaker!"](#)

XForms Model

← Prev Next →

Versão 1 – Documento em Atualização

The XForms input data can be described in two parts:

1) **The XForm model (to describe the data and the logic):** The XForms model is used to define what the form is, what it should do and what the data it should contain.

2) **The XForm user interface (to display and input the data):** The XForms user interface is used to define the input field and how they should be displayed.

The XForms Model

The XForm model is used to describe the data. It is an instance of an XML document. It defines the data model inside the <model> element:

The XForm model is used to describe the data. It is an instance of an XML document. It defines the data model inside the <model> element:

For Example:

```
<model>
<instance>
<person>
  <firstname/>
  <lastname/>
</person>
</instance>
<submission id="form1"
action="submit.asp"
method="get"/>
</model>
```



XForm Namespaces

The XForms Namespaces specifies the different instances which are used in the XForms forms.

The `<instance>` element:

The `<instance>` element is used to define the input data which has been collected. XForms is always collecting data for an XML document. The `<instance>` element in the XForms model defines the XML document.

See how the XML document in form is collecting data for looks like:

```
<person>
  <firstname/>
  <lastname/>
</person>
```