**XML Interview Questions with Answers:**

**1. What is XML?**

XML is the Extensible Markup Language. It improves the functionality  
of the Web by letting you identify your information in a more accurate,  
flexible, and adaptable way. It is extensible because it is not  
a fixed format like HTML (which is a single, predefined markup language).  
Instead, XML is actually a meta language—a language for describing  
other languages—which lets you design your own markup languages  
for limitless different types of documents. XML can do this because  
it’s written in SGML, the international standard meta language for  
text document markup (ISO 8879).

**2. What is a markup language?**

A markup language is a set of words and symbols for describing  
the identity of pieces of a document (for example ‘this is  
a paragraph’, ‘this is a heading’, ‘this  
is a list’, ‘this is the caption of this figure’,  
etc). Programs can use this with a style sheet to create output  
for screen, print, audio, video, Braille, etc.   
  
Some markup languages (eg those used in word processors) only describe  
appearances (’this is italics’, ‘this is bold’),  
but this method can only be used for display, and is not normally  
re-usable for anything else.

**3. Where should I use XML?**

Its goal is to enable generic SGML to be served, received, and  
processed on the Web in the way that is now possible with HTML.  
XML has been designed for ease of implementation and for interoperability  
with both SGML and HTML.   
Despite early attempts, browsers never allowed other SGML, only  
HTML (although there were plugins), and they allowed it (even encouraged  
it) to be corrupted or broken, which held development back for over  
a decade by making it impossible to program for it reliably. XML  
fixes that by making it compulsory to stick to the rules, and by  
making the rules much simpler than SGML.

But XML is not just for Web pages: in fact it’s very rarely used  
for Web pages on its own because browsers still don’t provide reliable  
support for formatting and transforming it. Common uses for XML  
include:   
Information identification because you can define your own markup,  
you can define meaningful names for all your information items.  
Information storage because XML is portable and non-proprietary,  
it can be used to store textual information across any platform.  
Because it is backed by an international standard, it will remain  
accessible and processable as a data format. Information structure  
  
XML can therefore be used to store and identify any kind of (hierarchical)  
information structure, especially for long, deep, or complex document  
sets or data sources, making it ideal for an information-management  
back-end to serving the Web. This is its most common Web application,  
with a transformation system to serve it as HTML until such time  
as browsers are able to handle XML consistently. Publishing the  
original goal of XML as defined in the quotation at the start of  
this section. Combining the three previous topics (identity, storage,  
structure) means it is possible to get all the benefits of robust  
document management and control (with XML) and publish to the Web  
(as HTML) as well as to paper (as PDF) and to other formats (eg  
Braille, Audio, etc) from a single source document by using the  
appropriate stylesheets. Messaging and data transfer XML is also  
very heavily used for enclosing or encapsulating information in  
order to pass it between different computing systems which would  
otherwise be unable to communicate. By providing a lingua franca  
for data identity and structure, it provides a common envelope for  
inter-process communication (messaging). Web services Building on  
all of these, as well as its use in browsers, machine-processable  
data can be exchanged between consenting systems, where before it  
was only comprehensible by humans (HTML). Weather services, e-commerce  
sites, blog newsfeeds, AJaX sites, and thousands of other data-exchange  
services use XML for data management and transmission, and the web  
browser for display and interaction.

**4. Why is XML such an important development?**

It removes two constraints which were holding back Web developments:  
1. dependence on a single, inflexible document type (HTML) which  
was being much abused for tasks it was never designed for;

2. the complexity of full SGML, whose syntax allows many powerful  
but hard-to-program options.  
XML allows the flexible development of user-defined document types.  
It provides a robust, non-proprietary, persistent, and verifiable  
file format for the storage and transmission of text and data both  
on and off the Web; and it removes the more complex options of SGML,  
making it easier to program for.

**5. Describe the differences between XML and HTML.**

It’s amazing how many developers claim to be proficient programming  
with XML, yet do not understand the basic differences between XML  
and HTML. Anyone with a fundamental grasp of XML should be able  
describe some of the main differences outlined in the table below.

**XML**User definable tags

Content driven  
End tags required for well formed documents  
Quotes required around attributes values  
Slash required in empty tags

**HTML**Defined set of tags designed for web display

Format driven  
End tags not required  
Quotes not required  
Slash not required

**6. Describe the role that XSL can play when dynamically  
generating HTML pages from a relational database.**

Even if candidates have never participated in a project involving  
this type of architecture, they should recognize it as one of the  
common uses of XML. Querying a database and then formatting the  
result set so that it can be validated as an XML document allows  
developers to translate the data into an HTML table using XSLT rules.  
Consequently, the format of the resulting HTML table can be modified  
without changing the database query or application code since the  
document rendering logic is isolated to the XSLT rules.

**7. What is SGML?**

SGML is the Standard Generalized Markup Language (ISO 8879:1986),  
the international standard for defining descriptions of the structure  
of different types of electronic document. There is an SGML FAQ  
from David Megginson at http://math.albany.edu:8800/hm/sgml/cts-faq.htmlFAQ;  
and Robin Cover’s SGML Web pages are at http://www.oasis-open.org/cover/general.html.  
For a little light relief, try Joe English’s ‘Not the SGML  
FAQ’ at http://www.flightlab.com/~joe/sgml/faq-not.txtFAQ.  
  
SGML is very large, powerful, and complex. It has been in heavy  
industrial and commercial use for nearly two decades, and there  
is a significant body of expertise and software to go with it.   
XML is a lightweight cut-down version of SGML which keeps enough  
of its functionality to make it useful but removes all the optional  
features which made SGML too complex to program for in a Web environment.

**8. Aren’t XML, SGML, and HTML all the same thing?**

Not quite; SGML is the mother tongue, and has been used for describing  
thousands of different document types in many fields of human activity,  
from transcriptions of ancient Irish manuscripts to the technical  
documentation for stealth bombers, and from patients’ clinical records  
to musical notation. SGML is very large and complex, however, and  
probably overkill for most common office desktop applications.

XML is an abbreviated version of SGML, to make it easier to use  
over the Web, easier for you to define your own document types,  
and easier for programmers to write programs to handle them. It  
omits all the complex and less-used options of SGML in return for  
the benefits of being easier to write applications for, easier to  
understand, and more suited to delivery and interoperability over  
the Web. But it is still SGML, and XML files may still be processed  
in the same way as any other SGML file (see the question on XML  
software).   
HTML is just one of many SGML or XML applications—the one  
most frequently used on the Web.   
Technical readers may find it more useful to think of XML as being  
SGML– rather than HTML++.

**9. Who is responsible for XML?**

XML is a project of the World Wide Web Consortium (W3C), and the  
development of the specification is supervised by an XML Working  
Group. A Special Interest Group of co-opted contributors and experts  
from various fields contributed comments and reviews by email.   
XML is a public format: it is not a proprietary development of any  
company, although the membership of the WG and the SIG represented  
companies as well as research and academic institutions. The v1.0  
specification was accepted by the W3C as a Recommendation on Feb  
10, 1998.

**10. Why is XML such an important development?**

It removes two constraints which were holding back Web developments:  
  
1. dependence on a single, inflexible document type (HTML) which  
was being much abused for tasks it was never designed for;  
2. the complexity of full question A.4, SGML, whose syntax allows  
many powerful but hard-to-program options.   
XML allows the flexible development of user-defined document types.  
It provides a robust, non-proprietary, persistent, and verifiable  
file format for the storage and transmission of text and data both  
on and off the Web; and it removes the more complex options of SGML,  
making it easier to program for.