

XML Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on XML will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [XML Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in XML category. To ensure quality, each submission is checked by our team, before it becomes live. This [XML Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



XML Interview Questions And Answers Guide.

Question - 1:

Is XML case sensitive?

Ans:

Yes, XML is a case sensitive language

[View All Answers](#)

Question - 2:

Tell me can we use graphics in XML?

Ans:

Yes, graphics can be stored in XML file by using XLink and XPointer. It supports graphics like GIF, JPG, TIFF, PNG, CGM, EPS, SVG.

[View All Answers](#)

Question - 3:

Which parts of an XML document are case-sensitive?

Ans:

All of it, both markup and text. This is significantly different from HTML and most other SGML applications. It was done to allow markup in non-Latin-alphabet languages, and to obviate problems with case-folding in writing systems which are caseless.

* Element type names are case-sensitive: you must follow whatever combination of upper- or lower-case you use to define them (either by first usage or in a DTD or Schema). So you can't say `<BODY>?</body>`: upper- and lower-case must match; thus ``, ``, and `` are three different element types;

* For well-formed XML documents with no DTD, the first occurrence of an element type name defines the casing;

* Attribute names are also case-sensitive, for example the two width attributes in `<PIC width="7in"/>` and `<PIC WIDTH="6in"/>` (if they occurred in the same file) are separate attributes, because of the different case of width and WIDTH;

* Attribute values are also case-sensitive. CDATA values (eg `Url="MyFile.SGML"`) always have been, but NAME types (ID and IDREF attributes, and token list attributes) are now case-sensitive as well;

* All general and parameter entity names (eg `Ã•`), and your data content (text), are case-sensitive as always.

[View All Answers](#)

Question - 4:

How does XML handle white-space in my documents?

Ans:

All white-space, including linebreaks, TAB characters, and normal spaces, even between ?structural? elements where no text can ever appear, is passed by the parser unchanged to the application (browser, formatter, viewer, converter, etc), identifying the context in which the white-space was found (element content, data content, or mixed content, if this information is available to the parser, eg from a DTD or Schema). This means it is the application's responsibility to decide what to do with such space, not the parser's:

* insignificant white-space between structural elements (space which occurs where only element content is allowed, ie between other elements, where text data never occurs) will get passed to the application (in SGML this white-space gets suppressed, which is why you can put all that extra space in HTML documents and not worry about it)

* significant white-space (space which occurs within elements which can contain text and markup mixed together, usually mixed content or PCDATA) will still get passed to the application exactly as under SGML. It is the application's responsibility to handle it correctly.

The parser must inform the application that white-space has occurred in element content, if it can detect it. (Users of SGML will recognize that this information is not in the ESIS, but it is in the Grove.)

[View All Answers](#)

Question - 5:

What does an XML document actually look like (inside)?

Ans:

The basic structure of XML is similar to other applications of SGML, including HTML. The basic components can be seen in the following examples. An XML document starts with a Prolog:

1. The XML Declaration



which specifies that this is an XML document;

2. Optionally a Document Type Declaration

which identifies the type of document and says where the Document Type Description (DTD) is stored;

The Prolog is followed by the document instance:

1. A root element, which is the outermost (top level) element (start-tag plus end-tag) which encloses everything else: in the examples below the root elements are conversation and titlepage;

2. A structured mix of descriptive or prescriptive elements enclosing the character data content (text), and optionally any attributes (?name=value? pairs) inside some start-tags.

XML documents can be very simple, with straightforward nested markup of your own design:

[View All Answers](#)

Question - 6:

Do I have to know HTML or SGML before I learn XML?

Ans:

No, although it's useful because a lot of XML terminology and practice derives from two decades' experience of SGML.

Be aware that ?knowing HTML? is not the same as ?understanding SGML?. Although HTML was written as an SGML application, browsers ignore most of it (which is why so many useful things don't work), so just because something is done a certain way in HTML browsers does not mean it's correct, least of all in XML.

[View All Answers](#)

Question - 7:

Does XML replace HTML?

Ans:

No. XML itself does not replace HTML. Instead, it provides an alternative which allows you to define your own set of markup elements. HTML is expected to remain in common use for some time to come, and the current version of HTML is in XML syntax. XML is designed to make the writing of DTDs much simpler than with full SGML. (See the question on DTDs for what one is and why you might want one.)

[View All Answers](#)

Question - 8:

What is the difference between XML and C or C++ or Java?

Ans:

C and C++ (and other languages like FORTRAN, or Pascal, or Visual Basic, or Java or hundreds more) are programming languages with which you specify calculations, actions, and decisions to be carried out in order:

```
mod curconfig[if left(date,6) = "01-Apr",
t.put "April google!";
f.put days("31102005","DDMMYYYY") -
days(sdate,"DDMMYYYY")
" more shopping days to Samhain"];
```

XML is a markup specification language with which you can design ways of describing information (text or data), usually for storage, transmission, or processing by a program. It says nothing about what you should do with the data (although your choice of element names may hint at what they are for):

```
<part num="DA42" models="LS AR DF HG KJ"
update="2001-11-22">
<name>Camshaft end bearing retention circlip</name>
<image drawing="RR98-dh37" type="SVG" x="476"
y="226"/> <maker id="RQ778">Ringtown Fasteners Ltd</maker>
<notes>Angle-nosed insertion tool <tool
id="GH25"/> is required for the removal
and replacement of this part.</notes>
</part>
```

On its own, an SGML or XML file (including HTML) doesn't do anything. It's a data format which just sits there until you run a program which

[View All Answers](#)

Question - 9:

When constructing an XML DTD, how do you create an external entity reference in an attribute value?

Ans:

Every interview session should have at least one trick question. Although possible when using SGML, XML DTDs don't support defining external entity references in attribute values. It's more important for the candidate to respond to this question in a logical way than than the candidate know the somewhat obscure answer.

[View All Answers](#)

Question - 10:

Using XSLT, how would you extract a specific attribute from an element in an XML document?

Ans:

Successful candidates should recognize this as one of the most basic applications of XSLT. If they are not able to construct a reply similar to the example below, they should at least be able to identify the components necessary for this operation: xsl:template to match the appropriate XML element, xsl:value-of to select the attribute value, and the optional xsl:apply-templates to continue processing the document.

Extract Attributes from XML Data

Example 1.

```
<xsl:template match="element-name">
Attribute Value:
<xsl:value-of select="@attribute"/>
<xsl:apply-templates/>
</xsl:template>
```



[View All Answers](#)

Question - 11:

Give some examples of XML DTDs or schemas that you have worked with?

Ans:

Although XML does not require data to be validated against a DTD, many of the benefits of using the technology are derived from being able to validate XML documents against business or technical architecture rules. Polling for the list of DTDs that developers have worked with provides insight to their general exposure to the technology. The ideal candidate will have knowledge of several of the commonly used DTDs such as FpML, DocBook, HRML, and RDF, as well as experience designing a custom DTD for a particular project where no standard existed.

[View All Answers](#)

Question - 12:

Can you walk us through the steps necessary to parse XML documents?

Ans:

Superficially, this is a fairly basic question. However, the point is not to determine whether candidates understand the concept of a parser but rather have them walk through the process of parsing XML documents step-by-step. Determining whether a non-validating or validating parser is needed, choosing the appropriate parser, and handling errors are all important aspects to this process that should be included in the candidate's response.

[View All Answers](#)

Question - 13:

Why should I use XML?

Ans:

Here are a few reasons for using XML (in no particular order). Not all of these will apply to your own requirements, and you may have additional reasons not mentioned here (if so, please let the editor of the FAQ know!).

* XML can be used to describe and identify information accurately and unambiguously, in a way that computers can be programmed to understand? (well, at least manipulate as if they could understand).

* XML allows documents which are all the same type to be created consistently and without structural errors, because it provides a standardised way of describing, controlling, or allowing/disallowing particular types of document structure. [Note that this has absolutely nothing whatever to do with formatting, appearance, or the actual text content of your documents, only the structure of them.]

* XML provides a robust and durable format for information storage and transmission. Robust because it is based on a proven standard, and can thus be tested and verified; durable because it uses plain-text file formats which will outlast proprietary binary ones.

[View All Answers](#)

Question - 14:

Why not just carry on extending HTML?

Ans:

HTML was already overburdened with dozens of interesting but incompatible inventions from different manufacturers, because it provides only one way of describing your information.

XML allows groups of people or organizations to question C.13, create their own customized markup applications for exchanging information in their domain (music, chemistry, electronics, hill-walking, finance, surfing, petroleum geology, linguistics, cooking, knitting, stellar cartography, history, engineering, rabbit-keeping, question C.19, mathematics, genealogy, etc).

HTML is now well beyond the limit of its usefulness as a way of describing information, and while it will continue to play an important role for the content it currently represents, many new applications require a more robust and flexible infrastructure.

[View All Answers](#)

Question - 15:

What is SOAP and how does it relate to XML?

Ans:

The Simple Object Access Protocol (SOAP) uses XML to define a protocol for the exchange of information in distributed computing environments. SOAP consists of three components: an envelope, a set of encoding rules, and a convention for representing remote procedure calls. Unless experience with SOAP is a direct requirement for the open position, knowing the specifics of the protocol, or how it can be used in conjunction with HTTP, is not as important as identifying it as a natural application of XML.

[View All Answers](#)

Question - 16:

What is DOM and how does it relate to XML?

Ans:

The Document Object Model (DOM) is an interface specification maintained by the W3C DOM Workgroup that defines an application independent mechanism to access, parse, or update XML data. In simple terms it is a hierarchical model that allows developers to manipulate XML documents easily Any developer that has worked extensively with XML should be able to discuss the concept and use of DOM objects freely. Additionally, it is not unreasonable to expect advanced candidates to thoroughly understand its internal workings and be able to explain how DOM differs from an event-based interface like SAX.

[View All Answers](#)

Question - 17:

Give a few examples of types of applications that can benefit from using XML?

Ans:



There are literally thousands of applications that can benefit from XML technologies. The point of this question is not to have the candidate rattle off a laundry list of projects that they have worked on, but, rather, to allow the candidate to explain the rationale for choosing XML by citing a few real world examples. For instance, one appropriate answer is that XML allows content management systems to store documents independently of their format, which thereby reduces data redundancy. Another answer relates to B2B exchanges or supply chain management systems. In these instances, XML provides a mechanism for multiple companies to exchange data according to an agreed upon set of rules. A third common response involves wireless applications that require WML to render data on hand held devices.

[View All Answers](#)

Question - 18:

Why is XML such an important development?

Ans:

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.

[View All Answers](#)

Question - 19:

Who is responsible for XML?

Ans:

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.

[View All Answers](#)

Question - 20:

Aren't XML, SGML, and HTML all the same thing?

Ans:

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++.

[View All Answers](#)

Question - 21:

What is SGML?

Ans:

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.html>; 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.txt>.

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.

[View All Answers](#)

Question - 22:

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

Ans:

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.

[View All Answers](#)

Question - 23:

Describe the differences between XML and HTML?

Ans:

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 to describe some of the main differences outlined in the table below.

Differences Between XML and HTML

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

[View All Answers](#)

Question - 24:

Why is XML such an important development?

Ans:

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.

[View All Answers](#)

Question - 25:

Where should I use XML?

Ans:

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.

[View All Answers](#)

Question - 26:

What is a markup language?

Ans:

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 stylesheet to create output for screen, print, audio, video, Braille, etc.

Some markup languages (eg those used in wordprocessors) 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.

[View All Answers](#)

Question - 27:

What is XML?

Ans:

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 metalanguage 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 metalanguage for text document markup (ISO 8879).

[View All Answers](#)

Computer Programming Most Popular & Related Interview Guides

- 1 : [Python Interview Questions and Answers.](#)
- 2 : [OOP Interview Questions and Answers.](#)
- 3 : [Software engineering Interview Questions and Answers.](#)
- 4 : [PHP Interview Questions and Answers.](#)
- 5 : [VBA \(Visual Basic for Applications\) Interview Questions and Answers.](#)
- 6 : [Visual Basic \(VB\) Interview Questions and Answers.](#)
- 7 : [Node.js Interview Questions and Answers.](#)
- 8 : [CMMI Interview Questions and Answers.](#)
- 9 : [Microsoft Foundation Class \(MFC\) Interview Questions and Answers.](#)
- 10 : [Lotus Notes Interview Questions and Answers.](#)

Follow us on FaceBook

www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

<https://InterviewQuestionsAnswers.ORG/>
support@InterviewQuestionsAnswers.ORG