XML Interview Questions with Answers:

21. What is the difference between XML and C or C++ or

Java? Updated

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 googlel!”,

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):

&lt;part num=”DA42″ models=”LS AR DF HG KJ”

update=”2001-11-22″&gt;

&lt;name&gt;Camshaft end bearing retention circlip&lt;/name&gt;

&lt;image drawing=”RR98-dh37″ type=”SVG” x=”476″

y=”226″/&gt; &lt;maker id=”RQ778″&gt;Ringtown

Fasteners Ltd&lt;/maker&gt;

&lt;notes&gt;Angle-nosed insertion tool &lt;tool

id=”GH25″/&gt; is required for the removal

and replacement of this part.&lt;/notes&gt;

&lt;/part&gt;

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 does something with it.

22. Does XML replace HTML?

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.)

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

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.

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

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:

&lt;?xml version=”1.0″ standalone=”yes”?&gt;

&lt;conversation&gt;&lt;br&gt;

&lt;greeting&gt;Hello, world!&lt;/greeting&gt;

&lt;response&gt;Stop the planet, I want to get

off!&lt;/response&gt;

&lt;/conversation&gt;

Or they can be more complicated, with a Schema or question C.11,

Document Type Description (DTD) or internal subset (local DTD changes

in [square brackets]), and an arbitrarily complex nested structure:

&lt;?xml version=”1.0″ encoding=”iso-8859-1″?&gt;

&lt;!DOCTYPE titlepage

SYSTEM “http://www.google.bar/dtds/typo.dtd”

[&lt;!ENTITY % active.links “INCLUDE”&gt;]&gt;

&lt;titlepage id=”BG12273624″&gt;

&lt;white-space type=”vertical” amount=”36″/&gt;

&lt;title font=”Baskerville” alignment=”centered”

size=”24/30″&gt;Hello, world!&lt;/title&gt;

&lt;white-space type=”vertical” amount=”12″/&gt;

&lt;!– In some copies the following

decoration is hand-colored, presumably

by the author –&gt;

&lt;image location=”http://www.google.bar/fleuron.eps”

type=”URI” alignment=”centered”/&gt;

&lt;white-space type=”vertical” amount=”24″/&gt;

&lt;author font=”Baskerville” size=”18/22″

style=”italic”&gt;Vitam capias&lt;/author&gt;

&lt;white-space type=”vertical” role=”filler”/&gt;

&lt;/titlepage&gt;

Or they can be anywhere between: a lot will depend on how you want

to define your document type (or whose you use) and what it will

be used for. Database-generated or program-generated XML documents

used in e-commerce is usually unformatted (not for human reading)

and may use very long names or values, with multiple redundancy

and sometimes no character data content at all, just values in attributes:

&lt;?xml version=”1.0″?&gt; &lt;ORDER-UPDATE AUTHMD5=”4baf7d7cff5faa3ce67acf66ccda8248″

ORDER-UPDATE-ISSUE=”193E22C2-EAF3-11D9-9736-CAFC705A30B3″

ORDER-UPDATE-DATE=”2005-07-01T15:34:22.46″ ORDER-UPDATE-DESTINATION=”6B197E02-EAF3-11D9-85D5-997710D9978F”

ORDER-UPDATE-ORDERNO=”8316ADEA-EAF3-11D9-9955-D289ECBC99F3″&gt;

&lt;ORDER-UPDATE-DELTA-MODIFICATION-DETAIL ORDER-UPDATE-ID=”BAC352437484″&gt;

&lt;ORDER-UPDATE-DELTA-MODIFICATION-VALUE ORDER-UPDATE-ITEM=”56″

ORDER-UPDATE-QUANTITY=”2000″/&gt;

&lt;/ORDER-UPDATE-DELTA-MODIFICATION-DETAIL&gt;

&lt;/ORDER-UPDATE&gt;

25. How does XML handle white-space in my documents?

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.)

&lt;chapter&gt;

&lt;title&gt;

My title for

Chapter 1.

&lt;/title&gt;

&lt;para&gt;

text

&lt;/para&gt;

&lt;/chapter&gt;

In the example above, the application will receive all the pretty-printing

linebreaks, TABs, and spaces between the elements as well as those

embedded in the chapter title. It is the function of the application,

not the parser, to decide which type of white-space to discard and

which to retain. Many XML applications have configurable options

to allow programmers or users to control how such white-space is

handled.

26. Which parts of an XML document are case-sensitive?

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 &lt;BODY&gt;…&lt;/body&gt;:

upper- and lower-case must match; thus &lt;Img/&gt;, &lt;IMG/&gt;,

and &lt;img/&gt; 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 &lt;PIC width=”7in”/&gt; and &lt;PIC WIDTH=”6in”/&gt;

(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 A), and your

data content (text), are case-sensitive as always.

27. How can I make my existing HTML files work in XML?

Either convert them to conform to some new document type (with

or without a DTD or Schema) and write a stylesheet to go with them;

or edit them to conform to XHTML. It is necessary to convert existing

HTML files because XML does not permit end-tag minimisation (missing

, etc), unquoted attribute values, and a number of other SGML shortcuts

which have been normal in most HTML DTDs. However, many HTML authoring

tools already produce almost (but not quite) well-formed XML.

You may be able to convert HTML to XHTML using the Dave Raggett’s

HTML Tidy program, which can clean up some of the formatting mess

left behind by inadequate HTML editors, and even separate out some

of the formatting to a stylesheet, but there is usually still some

hand-editing to do.

28. Is there an XML version of HTML?

Yes, the W3C recommends using XHTML which is ‘a reformulation

of HTML 4 in XML 1.0′. This specification defines HTML as

an XML application, and provides three DTDs corresponding to the

ones defined by HTML 4.\* (Strict, Transitional, and Frameset). The

semantics of the elements and their attributes are as defined in

the W3C Recommendation for HTML 4. These semantics provide the foundation

for future extensibility of XHTML. Compatibility with existing HTML

browsers is possible by following a small set of guidelines (see

the W3C site).

29. If XML is just a subset of SGML, can I use XML files

directly with existing SGML tools?

Yes, provided you use up-to-date SGML software which knows about

the WebSGML Adaptations TC to ISO 8879 (the features needed to support

XML, such as the variant form for EMPTY elements; some aspects of

the SGML Declaration such as NAMECASE GENERAL NO; multiple attribute

token list declarations, etc).

An alternative is to use an SGML DTD to let you create a fully-normalised

SGML file, but one which does not use empty elements; and then remove

the DocType Declaration so it becomes a well-formed DTDless XML

file. Most SGML tools now handle XML files well, and provide an

option switch between the two standards.

30. Can XML use non-Latin characters?

Yes, the XML Specification explicitly says XML uses ISO 10646,

the international standard character repertoire which covers most

known languages. Unicode is an identical repertoire, and the two

standards track each other. The spec says (2.2): ‘All XML

processors must accept the UTF-8 and UTF-16 encodings of ISO 10646…’.

There is a Unicode FAQ at http://www.unicode.org/faq/FAQ.

UTF-8 is an encoding of Unicode into 8-bit characters: the first

128 are the same as ASCII, and higher-order characters are used

to encode anything else from Unicode into sequences of between 2

and 6 bytes. UTF-8 in its single-octet form is therefore the same

as ISO 646 IRV (ASCII), so you can continue to use ASCII for English

or other languages using the Latin alphabet without diacritics.

Note that UTF-8 is incompatible with ISO 8859-1 (ISO Latin-1) after

code point 127 decimal (the end of ASCII).

UTF-16 is an encoding of Unicode into 16-bit characters, which lets

it represent 16 planes. UTF-16 is incompatible with ASCII because

it uses two 8-bit bytes per character (four bytes above U+FFFF).