XML DOM 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 DOM 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 DOM 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 DOM category. To ensure quality, each submission is checked by our team, before it becomes live. This XML DOM 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 DOM Interview Questions And Answers Guide.

Question - 1:

Where the Document Object Model (DOM) is used?

Ans:

Document Object Model (DOM) is used to query, traverse and manipulate documents like XML or HTML documents. DOM is best suited where the document must be accessed repeatedly or out of sequence order. DOM allows accessing the contents of a web page. It also allows dealing with events that allows capturing and responding to user's actions. There are different levels of DOM standards depending on the compatibility of the browsers.

View All Answers

Question - 2:

Define DOM?

Ans:

DOM is a platform independent, World Wide Web Consortium (W3C) standard form of representation of structured documents as an object-oriented model. It is an application programming interface so as to access HTML and XML documents.

View All Answers

Question - 3:

Explain the difference between DOM and SAX?

Ans:

SAX parser works incrementally and generates events that are passed to the application. DOM parser reads the whole XML document and returns a DOM tree representation of xml document
In DOM the xml file is arranged as a tree and backward and forward search is possible In SAX traversing in any direction is not possible as Top to bottom approach is used.
SAX is essentially an API for reading XML, and not writing it. DOM allows you to read and write.

View All Answers

Question - 4:

List the features of DOM?

Ans:

DOM is Document Object Model. It is used to read data from a XML document. It is more commonly used in applications where data in the document needs to be repeated accessed. DOM supports navigation in any direction. XML DOM is typically used for XML documents. The DOM defines the objects and properties of all document elements, and the methods (interface) to access them.

View All Answers

Question - 5:

Define HTML DOM?

Ans:

The HTML DOM API specializes and adds the functionality to relate to HTML documents and elements. It addresses the issues of backwards compatibility with the Level 0 of DOM and provides mechanisms for common and frequent operations on HTML documents.

View All Answers

Question - 6:

Define XMLHttpRequest Object?

Ans:

The XMLHttpRequest object is used to connect to the server through http. Scripts use it to do so programmatically.
The EventTarget interface needs to be implemented if an object implements the XMLHttpRequest interface. Also, an XMLHttpRequest() constructor needs to be provided by objects that implement the Window interface.

View All Answers

Question - 7:

What is CVS useful for?

Ans:

CVS is intended to handle source control for files in three major situations: 1. Multiple developers working on the same files.The major advantage of using CVS over the simpler tools like RCS or SCCS is that it allows multiple developers to work on the same sources at the same time.The shared Repository provides a rendezvous for committed sources that allows developers a fair amount of flexibility in how often to publish (via the "commit" command) changes or include work committed by others (via the "update" command). 2. Tracking a stream of releases from a source vendor.If you are making changes to sources distributed by someone else, the CVS feature, called the Vendor Branch, allows you to combine local modifications with repeated vendor releases.I have found this most useful when dealing with sources from three major classes of source vendor: a. Large companies who send you tapes full of the latest release (e.g. Unix OS vendors, database companies). b. Public Domain software which *always* requires work. c. Pseudo-Public sources which may require work. (e.g. GNU programs, X, CVS itself, etc.) 3. Branching development.Aside from the "Vendor Branch", there are three kinds of "branches in development" that CVS can support: a. Your working directory can be treated as a private branch. b. A Development branch can be shared by one or more developers. c. At release time, a branch is usually created for bug fixes. (See 1D.9 and Section 4C for more info on branches.) CVS's branch support is a bit primitive, but it was designed to allow you to create branches, work on them for while and merge them back into the main line of development.You should also be able to merge work performed on the main branch into the branch you are working on. Arbitrary sharing and merging between branches is not currently supported.

View All Answers

Question - 8:

How does CVS work?

Ans:

CVS saves its version-control information in RCS files stored in a directory hierarchy, called the Repository, which is separate from the user's working directory.Files in the Repository are stored in a format dictated by the RCS commands CVS uses to do much of its real work. RCS files are standard byte-stream files with an internal format described by keywords stored in the files themselves.To begin work, you execute a "checkout" command, handing it a module name or directory path (relative to the $CVSROOT variable)you want to work on. CVS copies the latest revision of each file in the specified module or directory out of the Repository and into a directory tree created in your current directory. You may specify a particular branch to work on by symbolic name if you don't want to work on the default (main or trunk) branch.You may then modify files in the new directory tree, build them into output files and test the results.When you want to make your changes available to other developers, you "commit" them back into the Repository.Other developers can check out the same files at the same time.To merge the committed work of others into your working files you use the "update" command. When your merged files build and test correctly, you may commit the merged result. This method is referred to as "copy-modify-merge", which does not require locks on the source files.At any time, usually at some milestone, you can "tag" the committed files, producing a symbolic name that can be handed to a future "checkout" command. A special form of "tag" produces a branch in development, as usually happens at "release" time. When you no longer plan to modify or refer to your local copy of the files, they can be removed.

View All Answers

Question - 9:

What is CVS *not* useful for?

Ans:

CVS is not a build system.Though the structure of your Repository and modules file interact with your build system (e.g. a tree of Makefiles),they are essentially independent.CVS does not dictate how you build anything. It merely stores files for retrieval in a tree structure you devise.CVS does not dictate how to use disk space in the checked out working directories. If you require your Makefiles or build procedures to know the relative positions of everything else,you wind up requiring the entire Repository to be checked out. That's simply bad planning.If you modularize your work, and construct a build system that will share files (via links, mounts, VPATH in Makefiles,etc.), you can arrange your disk usage however you like.But you have to remember that *any* such system is a lot of work to construct and maintain. CVS does not address the issues involved. You must use your brain and a ollection of other tools to provide a build scheme to match your plans.Of course, you should use CVS to maintain the tools created to support such a build system (scripts, Makefiles, etc).CVS is not a substitute for management.You and your project leaders are expected to plan what you are doing. Everyone involved must be aware of schedules, merge points, branch names, release dates and the range of procedures needed to build products. (If you produce it and someone else uses it, it is a product.) CVS can't cover for a failure to manage your project.CVS is an instrument for making sources dance to your tune.But you are the piper and the composer. No instrument plays itself or writes its own music.CVS is not a substitute for developer communication.When faced with conflicts within a single file, most developers manage to resolve them without too much effort.But a more general definition of "conflict" includes problems too difficult to solve without communication between developers.CVS cannot determine when simultaneous changes within a single file, or across a whole collection of files, will logically conflict with one another.Its concept of a "conflict" is purely textual, arising when two changes to the same base file are near enough to spook the merge command into dropping conflict markers into the merged file.CVS is not capable of figuring out distributed conflicts in program logic.For example, if you change the arguments to function X defined in file A and, at the same time, edit file B, adding new calls to function X using the old arguments.You are outside the realm of CVS's competence.Acquire the habit of reading specs and talking to your peers.CVS is not a configuration management system.CVS is a source control system. The phrase "configuration management" is a marketing term, not an industry-recognized set of functions.A true "configuration management system" would contain elements of the following: * Source control. * Dependency tracking. * Build systems (i.e. What to build and how to find things during a build. What is shared? What is local?) * Bug tracking. * Automated Testing procedures. * Release Engineering documentation and procedures. * Tape Construction. * Customer Installation. * A way for users to run different versions of the same software on the same host at the same time.

View All Answers

Question - 10:

How does CVS differ from RCS?

Ans:

CVS uses RCS to do much of its work and absolutely all the work of changing the underlying RCS files in the Repository. RCS comprises a set of programs designed to keep track of changes to individual files. Of course, it also allows you to refer to multiple files on the command line, but they are handled by iterating over individual files. There is no pretense of coordinated interaction among groups of files. CVS's main intent is to provide a set of grouping functions that allow you to treat a collection of RCS files as a single object. Of course, CVS also has to do a lot of iteration, but it tries its best to hide that it is doing so. In addition, CVS has some truly group-oriented facets, such as the modules file and the CVS administrative files that refer to a whole directory or module. One group aspect that can be a bit confusing is that a CVS branch is not the same as an RCS branch. To support a CVS branch, CVS uses "tags" (what RCS calls "symbols") and some local state, in addition to RCS branches. Other features offered by CVS that are not supported directly by RCS are 1. Automatic determination of the state of a file, (e.g. modified, up-to-date with the Repository, already tagged with the same string, etc.) which helps in limiting the amount of displayed text you have to wade through to figure out what changed and what to do next. 2. A copy-modify-merge scheme that avoids locking the files and allows simultaneous development on a single file. 3. Serialization of commits. CVS requires you to merge all changes committed (via "update") since you checked out your working copy of the file. Although it is still

possible to commit a file filled with old data, it is less likely than when using raw RCS.

View All Answers

Question - 11:

How does CVS differ from SCCS?

Ans:

SCCS is much closer to RCS than to CVS, so some of the previous entry applies.

View All Answers

Question - 12:

What is a Branch?

Ans:

In general, a branch is any mechanism that allows one or more developers to modify a file without affecting anyone other than those working on the same branch. There are four kinds of "branch" CVS can manage: 1. The Vendor Branch. A single vendor branch is supported. The "import" command takes a sequence of releases from a source code vendor (called a "vendor" even if no money is involved), placing them on a special "Vendor" branch. The Vendor branch is considered part of the "Main line" of development, though it must be merged into locally modified files on the RCS Main branch before the "import" is complete. See Section 3H ("import"). 2. Your Working directory. A checked-out working directory, can be treated like a private branch. No one but you can touch your files. You have complete control over when you include work committed by others. However, you can't commit or tag intermediate versions of your work. 3. A Development branch. A group of developers can share changes among the group, without affecting the Main line of development, by creating a branch. Only those who have checked-out the branch see the changes committed to that branch. This kind of branch is usually temporary, collapsing (i.e. merge and forget) into the Main line when the project requiring the branch is completed. You can also create a private branch of this type, allowing an individual to commit (and tag) intermediate revisions without changing the Main line. It should be managed exactly like a Development Branch -- collapsed into the Main line (or its parent branch, if that is not the Main Branch) and forgotten when the work is done. 4. A Release branch. At release time, a branch should be created marking what was released. Later, small changes (sometimes called "patches") can be made to the release without including everything else on the Main line of development. You avoid forcing the customer to accept new, possibly untested, features added since the release. This is also the way to correct bugs found during testing in an environment where other developers have continued to commit to the Main line while you are testing and packaging the release. Although the internal format of this type of branch (branch tag and RCS branches) is the same as in a development branch, its purpose and the way it is managed are different. The major difference is that a Release branch is normally Permanent. Once you let a release out the door to customers, or to the next stage of whatever process you are using, you should retain forever the branch marking that release. Since the branch is permanent, you cannot incorporate the branch fixes into the Main line by "collapsing" (merging and forgetting) the release branch. For large changes to many files on the release branch, you will have to perform a branch merge using "update -j <rev> -j <rev>". (See 4C.7) The most common way to merge small changes back into Main line development is to make the change in both places simultaneously. This is faster than trying to perform a selective merge.

View All Answers

Question - 13:

What is CVS for? What does it do for me?

Ans:

CVS is used to keep track of collections of files in a shared directory called "The Repository". Each collection of files can be given a "module" name, which is used to "checkout" that collection. After checkout, files can be modified (using your favorite editor), "committed" back into the Repository and compared against earlier revisions. Collections of files can be "tagged" with a symbolic name for later retrieval. You can add new files, remove files you no longer want, ask for information about sets of files in three different ways, produce patch "diffs" from a base revision and merge the committed changes of other developers into your working files.

View All Answers

Question - 14:

In your experience, who are the most important allies of SQA within an organization?

Ans:

SQA is a form of risk awareness, and is therefore potentially allied to any senior management with a risk management focus. Within some companies/industries (e.g. insurance), software risks are seen as having mainly financial consequences, and so the main ally might be the financial director. Within other companies/industries (e.g. retail), software risks are seen as having mainly customer service implications, and so the main allies may be in marketing roles. In one client, we had useful conversations with the Company Secretariat, because of the due diligence implications of some software risks. These conversations were triggered by Y2K issues, but ranged much more widely. In practice, SQA often fails to make these alliances, because it gets bogged down in obscure software technicalities and trivialities, which it is incapable of communicating effectively even to software engineers, let alone anybody else.

View All Answers

Question - 15:

How do I apply a label to an older version of a file?

Ans:

You must first pin the file at the older version (Show History, select the version of interest and click "Pin".) Then apply the label to the parent project. The label will be applied to the "tip" revision of the unpinned files, but to the older version of pinned files. Then Unpin the file. The label 'sticks' even after the pin is removed.

View All Answers

Question - 16:

How do I remove a label?

Ans:

This one is not intuitive. From within VSS Explorer, select the project, show history, check Labels Only, find the label of interest, click on Details, select the label text and delete it. Click Close, which will ask if you are sure you want to change the label. Click Yes and you're done.

View All Answers

Question - 17:

How do I find all changes between two dates?

Ans:

Here's how to get a list of everything that has changed since a particular date and time from the command line. In the command "a" or "p", refers to a.m. or p.m. respectively. And the ~ is used to indicate you want the history between the dates and times specified. Note that the later date and time must be specified first. -R makes it recursive. SS HISTORY $/mybranch -R -VD3/03/95;3:00p~3/03/95;9:00a

View All Answers

Question - 18:

How do I get with respect to a label?

Ans:

Need to use the '-v' switch with an 'L' after it to indicate a label (-R is for recursive): SS GET $project -VLmylabel -R

View All Answers

Question - 19:

How do I pin all the files within a project?

Ans:

You can do this using the command-line pin parameter and the wildcard character for the file name. For example: SS PIN $/MyProject/*.* -VLver1

View All Answers

Question - 20:

Where can I find the SSSCC API?

Ans:

* Microsoft discourages the use of the SSSCC API. Instead they propose you use the OLE Automation interface.
* However, if you still wish to use the API you can request the MSSCCI spec by writing to msscci@microsoft.com. They will send you the NDA you need to sign to get the spec, and then send you the spec.

View All Answers

Question - 21:

How do I select a database from the command line?

Ans:

You can set the SSDIR environment variable to point to the location of the database's SRCSAFE.INI file. • Chuck Kollars discusses the tradeoffs of a single vs multiple databases at his website.

View All Answers

Question - 22:

Does VSS OLE Automation support label comments?

Ans:

VSS 6.0 fully supports label comments. Label comments, however, are not available in VSS 5.0.

View All Answers

Question - 23:

Does VSS OLE Automation support administrative functions?

Ans:

VSS 6.0 supports some administrative functions.

View All Answers

Question - 24:

Is it possible to trap file deletions and rollbacks using an add-in/VSS Ole Automation?

Ans:

These events are not supported as of this posting by Rich Knox of Microsoft: Trapping Rollbacks/Deletions.

View All Answers

Question - 25:

How do I retrieve the comment from a specific version of a file?

Ans:

This requires that you iterate through each version of the file looking for the label you want. Once the correct version of the file is identified you can retrieve the comment for that version. Below is some sample VB code demonstrating this: For Each objVSSVersion In objVSSObject.Versions If objVSSVersion.Action = "Beta 1" Then MsgBox(objVSSVersion.LabelComment) End If Next

View All Answers

Question - 26:

What does CVS stand for? Can you describe it in one sentence?

Ans:

"CVS" is an acronym for the "Concurrent Versions System". CVS is a "Source Control" or "Revision Control" tool designed to keep track of source changes made by groups of developers working on the same files, allowing them to stay in sync with each other as each individual chooses.

View All Answers

Applications Programs Most Popular & Related Interview Guides

1 : **AutoCAD Interview Questions and Answers.**

2 : Microsoft Office Interview Questions and Answers.

3 : Microsoft Outlook Interview Questions and Answers.

4 : Microsoft Excel Interview Questions and Answers.

5 : MATLAB Interview Questions and Answers.

6 : WPF Interview Questions and Answers.

7 : OOAD Interview Questions and Answers.

8 : Quickbook Interview Questions and Answers.

9 : Microsoft Word Interview Questions and Answers.

10 : Bioinformatics 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