**Follow Up Interview Questions**

1. **What is the difference between Custom tags and ColdFusion Component (CFC) in ColdFusion?**

**(ANSWER) A component is a separate kind of CF file, saved with a .cfc extension, that allows functionality to be contained in a reusable way, but also in an object-oriented way. Aspects such as inheritance, scoping, type enforcement, etc. can be used when working with a component, although it is not necessary. A custom tag is a .cfm file that is stored in a specific ColdFusion directory and can be accessed in the tag syntax the same way other tags are referenced. A custom tag also provides a way to group together reusable functionality, but it lacks the added capabilities of a component.**

1. **What is the difference between JavaScript and jQuery?**

**(ANSWER) JavaScript is a programming language that is primarily used in the creation of web applications and is run in the browser on the end user’s (client’s) machine. JS pages are included as referenced files on an HTML page which are downloaded from the origin server and then executed. JS can also be used through other frameworks such as Node.js outside of a browser where the JS is interpreted locally on the host machine. jQuery is a library or framework that seeks to unify and simplify many common uses of JS in the browser, such as accessing the DOM, handling events, and using AJAX. jQuery is not a separate language.**

1. **What is a Framework and why would you use one?**

**(ANSWER) A framework is an extension of a programming language (a super set) that provides additional, prebuilt functionality on top of the existing language. This is generally a significant addition and can result in major functionality, capability, and syntax changes. Frameworks are usually built with specific goals in mind, but they (should) always strive to improve the development experience. With a library, you typically use the extra functionality specifically when needed, but with a framework, you are always operating within the framework’s extra layer. Frameworks can be useful when the framework addresses a problem you are having, and the framework itself is well-documented and supported. ColdBox or FW/1 seek to better organize CF code and to enforce structure, while also offering added capabilities. If you have a large piece of CF software, one of these (or other) frameworks might be useful to keep the source material more manageable, and speed up the development process.**

1. **What are some major frameworks?**

**(ANSWER) ColdBox (CF), FW/1 (CF), React.js (JS), Angular and AngularJS (JS), Vue.js (JS), Symfony (PHP), Laravel (PHP)**

1. **What is data normalization and why should you normalize data?**

**(ANSWER) Normalization is the process of defining a database’s structure such that it enforces a clear and concise structure on the data entered into it. This seeks to maintain the correctness of the data and the ability to clearly represent the relations between entries in a relational database. There are guidelines for achieving higher forms of normalization. Normalizing data is important to ensure that the database is able to properly store data and can do so in a way that avoids unnecessary redundancy or the possibility of data corruption.**

1. **What is SQL injection?**

**(ANSWER) SQL injection is a form of attacking a database via passing SQL or control statements in a dynamic query where values are expected, thus allowing the attacker to cause an unintended operation to be performed on the database. Because SQL statements are interpreted and are not pre-compiled (the statement is not known prior to execution), when the statement is interpreted, it is not possible to know what part of the statement is a direction to the database and which part is intended to be a value. For the statement: SELECT \* FROM users WHERE id = ‘*value*’. The expected input is a number that the application would enter. If the attacker can control the data in the *value* parameter (such as if it is passed as a URL parameter), then they can insert SQL such as ‘1 OR 1 = 1. Thus the database would return the entire user table because OR 1=1 is always a true statement.**

1. **How can you use ColdFusion to prevent SQL injection?**

**(ANSWER) Like (hopefully) all languages that incorporate database access, ColdFusion has the ability to bind parameters in a query. This essentially creates a slot or bucket in parts of the query that tells the database these areas do not contain SQL statements and should not be interpreted as such. This also has the benefit of allowing the database to cache and reuse these statements (not the result set) as they are technically always the same statement. SELECT \* FROM users WHERE id = [bucket] can be run with different values in the bucket but it is the same query to the database, whereas SELECT \* FROM users WHERE id = ‘*1*’, SELECT \* FROM users WHERE id = ‘*2*’, SELECT \* FROM users WHERE id = ‘*601*’ are all different queries because they are not exactly the same.**

1. **How would you handle being asked to work on a project that seems doomed to failure? (Incredibly short timeline, still emerging requirements, highly abbreviated or no planning and testing stages?)**

**(ANSWER) I would make it clear that the project seems flawed and elaborate on the specific failings so that it was clear to everyone involved that the project will have significant problems. I would recommend that the project be put on hold until these problems could be addressed. If that would not be feasible, I would work on the project to the best of my abilities.**

1. **What would you call your native programming Language?**

**(ANSWER) I would say ColdFusion or JavaScript.**

1. **What is the first version of ColdFusion you used?**

**(ANSWER) ColdFusion 2016.**

1. **cfscript…   
    What is it? Pros? Cons?  
    Do you like it? (Why/Why not)  
    Do you use it?**

**(ANSWER) cfscript is the script based way to write ColdFusion as opposed to cftags. cfscript is similar to other programming languages as opposed to cftags, and there are some things that you can do with cfscript that are not yet available in tags, such as marking a component as abstract. The downside is that it can be hard to find documentation for the script syntax. Try Googling coldfusion query, and you will get plenty of info about using the cfquery tag but not a lot about the queryExecute() function. I do use it and I think it would be generally preferable to tags since it is much less of a stylistic change from other languages that I would also be using.**

1. **What do you like about programming?**

**(ANSWER) It is solving a problem. It is approaching something logically. There are many ways to solve the problem, but you can learn better ways of doing it, so it has a fairly low barrier to entry and a high skill ceiling. In web development in particular, it is very easy to create something that you can instantly see without the need for any special tools; you can create an HTML page in Notepad and open it with a browser.**

1. **Which form field type does not pass a value to the action page by default?**

**(ANSWER) I believe that any form field type will pass its value when submitting the form, even hidden type fields. Perhaps in an instance where the form data is multipart such as uploading and using a file input would not send the file contents to the action page. (Assuming that the page is CF)**

1. **How can you prevent a form field from passing its value to the action page?**

**(ANSWER) Adding the disabled attribute to a form field prevents it from passing its value when the form submits. (Assuming that the page is CF)**

1. **Which type of validation is the most secure, client-side, manual server-side, or auto-generated server-side?**

**(ANSWER) Client-side validation is never secure because an attacker could potentially disable the validation, bypass it, or alter it. The purpose of client-side validation is purely to make the user experience better. It is more user friendly to inform the user that an input is invalid before they submit a form, rather than processing the form and then telling the user they need to fix something. Personally, I feel that it depends if manual or automatic server-side validation is more secure. Manual requires direct developer input and consideration, so the validation may be more specific to the field and specific instance, but it requires the developer to add validation everywhere, and that leaves the possibility of missing it. Automatic validation would (hopefully) not miss any instances where validation is needed, but may not perform sufficient or correct validation in every instance.**

1. **When selecting from a table that contained book titles, how could you return all the books that start with the word “The” in the title field?**

**(ANSWER) SELECT \* FROM books WHERE UPPER(title) LIKE ‘THE %’ (please note there is a single space between E and the %)**

1. **What is the difference between cfparam and cfset?**

**(ANSWER) cfset is used to create a variable, optionally locally scope it with var in a function, and to call functions. cfparam is used to create a variable, but it will check if the variable name already exists, can set a default value if it does not, and can validate other attributes about the value.**

1. **What is the hardest problem you have had to solve with ColdFusion and how did you solve it?**

**(ANSWER) With ColdFusion specifically, we had an application that delivered push notifications to iOS and Android. The process to send them via FCM for Android was straightforward. The process for iOS required more complex steps, and was originally handled by a small, 3rd party CF package. When Apple updated the APNS process, the old package did not work anymore and had received no updates to be compatible with the new process. I was unable to perform some of the necessary steps with CF. I had to utilize Java to perform the steps, with the Java code written in the ColdFusion and then called by CF code.**

**Another problem that dealt entirely with ColdFusion had to do with creating a report that provided a .csv file as output. There were other reports in the codebase like this already, so I referenced them to complete my task. The report had a large volume of data, and it took a long time for the report to be generated. This was being done with the tag syntax exclusively. I looked for ways to improve the performance. After testing, I found that the cffile tag is very inefficient. When using functions to manipulate a file, there are functions for opening the file, writing to the file, and closing the file. The cffile tag does not specify opening and closing, and there are no other tags to do this. It seems that the cffile tag opens and closes the file every time it is executed. The report just writes a line to the file in a loop. This meant that the file was constantly being opened and closed every time a line was written to it, which was the source of the significant increase in runtime. I changed my report to use the file manipulation functions, presented my findings to the team along with a simple test example to illustrate the disparity, and then was allowed to change the existing reports to the much faster approach.**