

Comprehensive ColdFusion

Course Summary

Description

In this Comprehensive ColdFusion training course, students will spend the first part of the week learning the basics of ColdFusion and the second part of the week learning advanced ColdFusion techniques.

Class is conducted in 2016 but students using versions 9, 10, 11, or 2016 may attend.

Objectives

After taking this course, students will be able to:

- Learn to get data from a database and display it on the page
- Learn to insert, Edit and Delete records from a database
- Learn to pass data from one page and receive it on another using HTML Forms
- Learn to create Dynamic Form Fields including Dynamic Select Lists
- Learn to build a Search Interface
- Learn to use Session Variables to provide Password Protection
- Learn to call ColdFusion Components (CFCs) and Custom Tags
- Learn to create and use complex data objects.
- Learn to write `<cfscript>` code.
- Learn to build and use Custom Tags.
- Learn to build and use ColdFusion Components (CFCs).
- Learn to handle errors properly.
- Learn to page through recordsets.
- Learn advanced form validation with regular expressions.

Topics

- Introduction to ColdFusion
- ColdFusion Basics
- Dynamic Data: Querying the Database
- Creating a Drill-down Interface
- Conditional Statements and Forms
- Quick Introduction to RSS and Ajax with `<cfform>`
- Search Interface
- Insert Records into the Database
- Edit (Update) Database Records
- Reusing Code
- Using Session Variables for Password Protection
- Tracking User Information with Cookies (optional)
- Sending email with `<cfmail>` (optional)
- Charting
- Complex Objects
- Components
- Application Framework
- Error Handling
- Reusing Code
- Regular Expressions (optional)

Audience

This course is designed for those wanting to learn both basic and advanced techniques for ColdFusion.

Prerequisites

Before taking this course, students are required to have experience in HTML. Experience in SQL, JavaScript, and CSS would be beneficial.

Duration

Five days

Comprehensive ColdFusion

Course Outline

- I. Introduction to ColdFusion**
 - A. Static pages vs. Dynamic pages
 - B. Server-side Web Technologies
 - C. Database Alphabet Soup: ODBC/JDBC, DSNs and SQL
 - D. Course Objectives and Prerequisites
 - E. Introduction to Course Project
- II. ColdFusion Basics**
 - A. Using CFML
 - 1. Code Editors
 - 2. Dreamweaver Interface
 - 3. Tag Updaters / Extensions
 - 4. Tag Editing Dialog Boxes
 - B. Embedded Tag Help
 - C. Using ColdFusion Comments
 - D. Creating Variables
 - 1. <cfset> and <cfoutput>
 - E. Variable Scope/Prefixes
 - F. Built-in ColdFusion Functions
 - 1. CF Function Reference (Dreamweaver specific)
 - 2. Online Reference (<http://livedocs.adobe.com>)
 - 3. Tip: (<http://cfquickdocs.com>)
 - G. Using <cfinclude> to reuse code
- III. Dynamic Data: Querying the Database**
 - A. Database Basics
 - 1. Introduction to SQL
 - 2. Select Statement
 - 3. Update Statement
 - 4. Insert Statement
 - 5. Delete Statement
 - 6. Data Source Name (DSN) / Remote Development Server (RDS)
 - 7. The DSN Used in This Class: movieList
 - B. Course Project Database
 - C. Querying the database with <cfquery>
 - 1. <cfdump> for Quick Output
 - D. Customizing Display with <cfoutput>
 - 1. Building Dynamic Tables
 - 2. Shortcut: <cftable>
 - 3. Shortcut: Dynamic Table Tool (in Dreamweaver)
- IV. Creating a Drill-down Interface**
 - A. Passing Variables between Pages
 - 1. "Two-page method"
 - B. Receive Variables through the URL
 - 1. Using a WHERE clause in a SQL Statement
 - 2. Protecting Against SQL Injection Attacks with <cfqueryparam>
- V. Conditional Statements and Forms**
 - A. Conditional Statements
 - 1. <cfif>, <cfelseif> and <cfelse>
 - 2. Comparison Operators
 - 3. Operator
 - 4. Description
 - 5. Using the isDefined() function
 - B. HTML Form Review
 - C. Self-Submitting Forms
 - 1. Using the <cfparam> tag to set a default value
 - 2. Quick Examination of a ColdFusion Framework (Methodology): Fusebox
 - D. Dynamically Populating Select Lists
 - 1. HTML Review: <select> tags
 - 2. Introducing the <cfform> Tag
 - 3. Shortcut: <cfselect>
- VI. Quick Introduction to RSS and Ajax with <cfform>**
 - A. Reading an RSS Feed
 - B. Introducing the <cfform> Tag
 - C. Building a Mask with <cfform>
 - D. Automatically Build a Dynamic Select List with <cfselect>
- VII. Search Interface**
 - A. Creating a Search Interface
 - 1. Using % and _ in a LIKE Statement

Comprehensive ColdFusion

Course Outline (cont'd)

VIII. Insert Records into the Database

- A. Inserting Records
 1. SQL Insert Statement
 2. Insert Record Step 2 - the Insert
 3. Shortcut: <cfinsert>
 4. Redirect Users with <cflocation>
- B. Automated Form Validation
 1. Client-side vs. Server-side validation: When does the validation code run?

IX. Edit (Update) Database Records

- A. Updating records
 1. SQL UPDATE Statement Review
 2. Passing the Primary Key
 3. Page 2 - View Current Data in a Form
 4. Page 3 - Perform the Update to the Database
 5. Shortcut: <cfupdate>
- B. Deleting Records
 1. Note on Deleting records
 2. Review of the SQL Delete Statement
 3. "Deleting" without deleting
 4. No <cfdelete>!

X. Reusing Code

- A. Reusing code with <cfinclude> and <cflocation>
- B. Custom Tags
 1. Where to store Custom Tags
 2. CALLER and ATTRIBUTE scopes
 3. Using <cfmodule>
 4. A Note about CFX tags
- C. User Defined Functions (UDFs) and ColdFusion Components (CFCs)
 1. Creating UDFs with the <cffunction> tag
 2. Using ColdFusion Components (CFCs)
- D. Creating Components
 1. Calling or Invoking Components
 2. Passing Arguments to Components
 3. The Component Code

4. The main CFML page
5. View a Component directly in the browser

XI. Using Session Variables for Password Protection

- A. Session Variables
 1. Using Application.cfc to Enable Session Management
 2. Creating Password Protected Pages
 3. Password-protect.cfm (the included page)
 4. Application.cfm
 5. Locking Session and Application variables with <cflock>

XII. Tracking User Information with Cookies (optional)

- A. Using Cookies to Track Users

XIII. Sending email with <cfmail> (optional)

- A. Using ColdFusion to Send Email
 1. <cfmail> Tag Syntax

XIV. Charting

- A. Course Introduction and <cfchart>
 1. Introduction
 2. Course Objectives
 3. Course Prerequisites
 4. Course Project
- B. Charting and Graphing Options
 1. Using <cfchart> vs. <cfchartseries>
 2. Caching Charts

XV. Complex Objects

- A. Manipulating Complex Data Objects (Lists, Arrays, Structures and Queries)
 1. Scalar Variables
 2. Lists
 3. Arrays
 4. Structures
- B. Looping with <cfloop>
 1. Syntax of <cfloop>

Comprehensive ColdFusion

Course Outline (cont'd)

2. Weeding Out the FIELDNAMES Variable
 3. Using <cfloop> for automatic query output
 4. Display Web-safe Colors using CFLOOP
- C. Queries of Queries

XVI. Components

- A. CFCs - Building ColdFusion Components
1. Benefits to using CFCs
 2. Creating Components
 3. Calling or Invoking Components
 4. Passing Arguments to Components
 5. The main CFML page
 6. Three Ways to Pass Arguments
 7. Where to save CFCs
 8. Self-documenting - View a Component directly in the browser
- B. Object-Oriented CFCs
1. Object-Oriented Principles
 2. How does use of a CFC compare to other OO languages?
 3. Inheritance
 4. Access control
 5. Exposing CFCs as Web Services

XVII. Application Framework

- A. Application Framework
1. Automatic Includes: Application.cfm and onRequestEnd.cfm
 2. New Application Framework: Application.cfc
- B. Using Application Variables

XVIII. Error Handling

- A. Errors and Exceptions
1. Standard Error Handling
 2. Application.cfm
- B. Structured Error Handling
1. <cftry> and <cfcatch>
 2. Transactions with <cftransaction> and <cftry>
 3. Classic <cftransaction> example

XIX. Reusing Code

- A. Reusing Code
1. Limited Capabilities of <cfscript>
 2. Syntax Rules
 3. Reserved Words
 4. Looping with <cfscript>
 5. GetTickCount() - testing the speed of your code
- B. Custom Tags
1. Where to store Custom Tags
 2. CALLER and ATTRIBUTE scopes
 3. Using <cfmodule>
- C. Stored Procedures
1. Benefits of Stored Procedures
- D. Showing portions of a Recordset
1. Paging through records

XX. Regular Expressions (optional)

- A. Advanced Validation with Regular Expressions (optional)
1. Special Symbols for Matching Specific Characters
 2. Specify the number of occurrences
 3. Placement of the pattern
 4. POSIX Syntax
 5. Perl Syntax