

Comprehensive ColdFusion Training

(5 Day)

Class Overview

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.

Course Benefits

- 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.

Class Outline

Introduction to ColdFusion

Static pages vs. Dynamic pages
Server-side Web Technologies
Database Alphabet Soup: ODBC/JDBC, DSNs and SQL
Course Objectives and Prerequisites
Introduction to Course Project

ColdFusion Basics

Using CFML
Code Editors

- Dreamweaver Interface
- Tag Updaters / Extensions
- Tag Editing Dialog Boxes
- Embedded Tag Help
- Using ColdFusion Comments
- Creating Variables
 - <cfset> and <cfoutput>
- Variable Scope/Prefixes
- Built-in ColdFusion Functions
 - CF Function Reference (Dreamweaver specific)
 - Online Reference (<http://livedocs.adobe.com>)
 - Tip: (<http://cfquickdocs.com>)
- Using <cfinclude> to reuse code

Dynamic Data: Querying the Database

- Database Basics
 - Introduction to SQL
 - Select Statement
 - Update Statement
 - Insert Statement
 - Delete Statement
 - Data Source Name (DSN) / Remote Development Server (RDS)
 - The DSN Used in This Class: movieList
- Course Project Database
- Querying the database with <cfquery>
 - <cfdump> for Quick Output
- Customizing Display with <cfoutput>
 - Building Dynamic Tables
 - Shortcut: <cftable>
 - Shortcut: Dynamic Table Tool (in Dreamweaver)

Creating a Drill-down Interface

- Passing Variables between Pages
 - "Two-page method"
- Receive Variables through the URL
 - Using a WHERE clause in a SQL Statement
 - Protecting Against SQL Injection Attacks with <cfqueryparam>

Conditional Statements and Forms

- Conditional Statements
 - <cfif>, <cfelseif> and <cfelse>
 - Comparison Operators
 - Operator
 - Description
 - Using the isDefined() function
- HTML Form Review
- Self-Submitting Forms
 - Using the <cfparam> tag to set a default value
 - Quick Examination of a ColdFusion Framework (Methodology): Fusebox
- Dynamically Populating Select Lists

HTML Review: <select> tags
Introducing the <cform> Tag
Shortcut: <cfselect>

Quick Introduction to RSS and Ajax with <cform>

Reading an RSS Feed
Introducing the <cform> Tag
Building a Mask with <cform>
Automatically Build a Dynamic Select List with <cfselect>

Search Interface

Creating a Search Interface
Using % and _ in a LIKE Statement

Insert Records into the Database

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

Edit (Update) Database Records

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

Reusing Code

Reusing code with <cfinclude> and <cflocation>
Custom Tags
Where to store Custom Tags
CALLER and ATTRIBUTE scopes
Using <cfmodule>
A Note about CFX tags
User Defined Functions (UDFs) and ColdFusion Components (CFCs)
Creating UDFs with the <cfunction> tag
Using ColdFusion Components (CFCs)
Creating Components
Calling or Invoking Components
Passing Arguments to Components
The Component Code

The main CFML page
View a Component directly in the browser

Using Session Variables for Password Protection

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

Tracking User Information with Cookies (optional)

Using Cookies to Track Users

Sending email with <cfmail> (optional)

Using ColdFusion to Send Email
<cfmail> Tag Syntax

Charting

Course Introduction and <cfchart>
Introduction
Course Objectives
Course Prerequisites
Course Project
Charting and Graphing Options
Using <cfchart> vs. <cfchartseries>
Caching Charts

Complex Objects

Manipulating Complex Data Objects (Lists, Arrays, Structures and Queries)
Scalar Variables
Lists
Arrays
Structures
Looping with <cfloop>
Syntax of <cfloop>
Weeding Out the FIELDNAMES Variable
Using <cfloop> for automatic query output
Display Web-safe Colors using CFLOOP
Queries of Queries

Components

CFCs - Building ColdFusion Components
Benefits to using CFCs
Creating Components
Calling or Invoking Components
Passing Arguments to Components
The main CFML page
Three Ways to Pass Arguments
Where to save CFCs

- Self-documenting - View a Component directly in the browser
- Object-Oriented CFCs
 - Object-Oriented Principles
 - How does use of a CFC compare to other OO languages?
 - Inheritance
 - Access control
 - Exposing CFCs as Web Services

Application Framework

- Application Framework
 - Automatic Includes: Application.cfm and onRequestEnd.cfm
 - New Application Framework: Application.cfc
 - Using Application Variables

Error Handling

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

Reusing Code

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

Regular Expressions (optional)

- Advanced Validation with Regular Expressions (optional)
 - Special Symbols for Matching Specific Characters
 - Specify the number of occurrences
 - Placement of the pattern
 - POSIX Syntax
 - Perl Syntax