



## CFML Basics

### Syntax Fundamentals

Tags	<p>CFML uses tags (similar to HTML) to define code blocks.</p> <p><b>Example:</b></p> <pre>&lt;cfoutput&gt;Hello, World!&lt;/cfoutput&gt;</pre>
Variables	<p>Variables are dynamically typed and declared using <code>&lt;cfset&gt;</code> or <code>&lt;cfparam&gt;</code>.</p> <p><b>Example:</b></p> <pre>&lt;cfset myVariable = "Some Value"&gt; &lt;cfparam name="anotherVariable" default="Default Value"&gt;</pre>
Comments	<p>Single-line and multi-line comments are supported.</p> <p><b>Example:</b></p> <pre>&lt;!-- This is a single-line comment --&gt; &lt;cfscript&gt; /*   This is a multi-line   comment. */ &lt;/cfscript&gt;</pre>
Output	<p>The <code>&lt;cfoutput&gt;</code> tag is used to display content.</p> <p><b>Example:</b></p> <pre>&lt;cfoutput&gt;The value of myVariable is: #myVariable#&lt;/cfoutput&gt;</pre>
CFScript	<p>CFScript allows you to write CFML code using a scripting syntax (similar to JavaScript).</p> <p><b>Example:</b></p> <pre>&lt;cfscript&gt;   name = "John";   writeOutput("Hello, " &amp; name); &lt;/cfscript&gt;</pre>
Data Types	<p>CFML supports various data types including String, Numeric, Boolean, Date, Array, and Struct.</p> <p><b>Example:</b></p> <pre>&lt;cfset myNumber = 123&gt; &lt;cfset myBoolean = true&gt; &lt;cfset myDate = now()&gt;</pre>

### Control Structures

### Variable Scopes

Application	<p>Variables available throughout the entire application.</p> <p><b>Example:</b></p> <pre>&lt;cfapplication name="MyApp" sessionmanagement="true"&gt; &lt;cfset application.myVar = "Application Value"&gt;</pre>
Session	<p>Variables specific to a user's session.</p> <p><b>Example:</b></p> <pre>&lt;cfset session.user_id = 123&gt;</pre>
Request	<p>Variables available during a single request.</p> <p><b>Example:</b></p> <pre>&lt;cfset request.pageTitle = "Home Page"&gt;</pre>
Server	<p>Variables available throughout the entire server.</p> <p><b>Example:</b></p> <pre>&lt;cfset server.db_connection = "DSN=MyDatabase"&gt;</pre>
Client	<p>Variables stored on the client's machine (usually via cookies).</p> <p><b>Example:</b></p> <pre>&lt;cfset client.username = "johndoe"&gt;</pre>
Local	<p>Variables available within a specific function or component.</p> <p><b>Example:</b></p> <pre>&lt;cffunction name="myFunc"&gt;   &lt;cfset var localVar = "Function Value"&gt; &lt;/cffunction&gt;</pre>

## Conditional Statements

<code>&lt;cfif&gt;</code>	Basic conditional statement.
<b>Example:</b>	<pre>&lt;cfif myVariable EQ "Some Value"&gt;   &lt;cfoutput&gt;Condition is true. &lt;/cfoutput&gt; &lt;cfelse&gt;   &lt;cfoutput&gt;Condition is false. &lt;/cfoutput&gt; &lt;/cfif&gt;</pre>
<code>&lt;cfelseif&gt;</code>	Used for multiple conditions.
<b>Example:</b>	<pre>&lt;cfif score GTE 90&gt;   &lt;cfoutput&gt;A&lt;/cfoutput&gt; &lt;cfelseif score GTE 80&gt;   &lt;cfoutput&gt;B&lt;/cfoutput&gt; &lt;cfelse&gt;   &lt;cfoutput&gt;C&lt;/cfoutput&gt; &lt;/cfif&gt;</pre>
<code>&lt;cfswitch&gt;</code>	Switch statement for multiple cases.
<b>Example:</b>	<pre>&lt;cfswitch expression="#myVariable#"   &lt;cfcase value="Value1"&gt;     &lt;cfoutput&gt;Case 1&lt;/cfoutput&gt;   &lt;/cfcase&gt;   &lt;cfcase value="Value2"&gt;     &lt;cfoutput&gt;Case 2&lt;/cfoutput&gt;   &lt;/cfcase&gt;   &lt;cfdefaultcase&gt;     &lt;cfoutput&gt;Default Case&lt;/cfoutput&gt;   &lt;/cfdefaultcase&gt; &lt;/cfswitch&gt;</pre>

## Looping Structures

<code>&lt;cfloop index="i" from="1" to="10"&gt;</code>	Numeric loop.
<b>Example:</b>	<pre>&lt;cfloop index="i" from="1" to="10"&gt;   &lt;cfoutput&gt;#i# &lt;/cfoutput&gt; &lt;/cfloop&gt;</pre>
<code>&lt;cfloop array="myArray" index="item"&gt;</code>	Looping through an array.
<b>Example:</b>	<pre>&lt;cfset myArray = ["A", "B", "C"]&gt; &lt;cfloop array="#myArray#" index="item"&gt;   &lt;cfoutput&gt;#item# &lt;/cfoutput&gt; &lt;/cfloop&gt;</pre>
<code>&lt;cfloop query="myQuery"&gt;</code>	Looping through a query result.
<b>Example:</b>	<pre>&lt;cfquery name="myQuery" datasource="myDSN"&gt;   SELECT id, name FROM users &lt;/cfquery&gt; &lt;cfloop query="myQuery"&gt;   &lt;cfoutput&gt;#myQuery.id# - #myQuery.name#&lt;br&gt; &lt;/cfoutput&gt; &lt;/cfloop&gt;</pre>
<code>&lt;cfloop condition="i LTE 10"&gt;</code>	Conditional loop.
<b>Example:</b>	<pre>&lt;cfset i = 1&gt; &lt;cfloop condition="i LTE 10"&gt;   &lt;cfoutput&gt;#i# &lt;/cfoutput&gt;   &lt;cfset i = i + 1&gt; &lt;/cfloop&gt;</pre>
<code>&lt;cfloop list="myList" index="item" delimiters=", "&gt;</code>	Looping through a list.
<b>Example:</b>	<pre>&lt;cfset myList = "A,B,C"&gt; &lt;cfloop list="#myList#" index="item" delimiters=", "&gt;   &lt;cfoutput&gt;#item# &lt;/cfoutput&gt; &lt;/cfloop&gt;</pre>

## Exception Handling

<code>&lt;cftry&gt;</code> and <code>&lt;cfcatch&gt;</code>	Exception handling block.
<b>Example:</b>	<pre>&lt;cftry&gt;   &lt;cfset result = 10 / 0&gt; &lt;cfcatch type="Any"&gt;   &lt;cfoutput&gt;Error: #cfcatch.message#&lt;/cfoutput&gt; &lt;/cfcatch&gt; &lt;/cftry&gt;</pre>

## Functions

## String Functions

<code>UCase(string)</code>	Converts a string to uppercase.  <b>Example:</b> <pre>&lt;cfset myString = "hello"&gt; &lt;cfoutput&gt;#UCase(myString)# &lt;/cfoutput&gt; &lt;!-- Output: HELLO --&gt;</pre>
<code>LCase(string)</code>	Converts a string to lowercase.  <b>Example:</b> <pre>&lt;cfset myString = "HELLO"&gt; &lt;cfoutput&gt;#LCase(myString)# &lt;/cfoutput&gt; &lt;!-- Output: hello --&gt;</pre>
<code>Left(string, count)</code>	Returns the leftmost characters of a string.  <b>Example:</b> <pre>&lt;cfset myString = "ColdFusion"&gt; &lt;cfoutput&gt;#Left(myString, 4)# &lt;/cfoutput&gt; &lt;!-- Output: Cold --&gt;</pre>
<code>Right(string, count)</code>	Returns the rightmost characters of a string.  <b>Example:</b> <pre>&lt;cfset myString = "ColdFusion"&gt; &lt;cfoutput&gt;#Right(myString, 6)#&lt;/cfoutput&gt; &lt;!-- Output: Fusion --&gt;</pre>
<code>Mid(string, start, count)</code>	Extracts a substring from a string.  <b>Example:</b> <pre>&lt;cfset myString = "ColdFusion"&gt; &lt;cfoutput&gt;#Mid(myString, 5, 4)#&lt;/cfoutput&gt; &lt;!-- Output: Fusi --&gt;</pre>
<code>Len(string)</code>	Returns the length of a string.  <b>Example:</b> <pre>&lt;cfset myString = "ColdFusion"&gt; &lt;cfoutput&gt;#Len(myString)# &lt;/cfoutput&gt; &lt;!-- Output: 10 --&gt;</pre>
<code>Find(substring, string)</code>	Returns the starting position of a substring within a string.  <b>Example:</b> <pre>&lt;cfset myString = "ColdFusion"&gt; &lt;cfoutput&gt;#Find("Fusion", myString)#&lt;/cfoutput&gt; &lt;!-- Output: 5 --&gt;</pre>

<code>Replace(string, oldSubstring, newSubString, scope)</code>	Replaces occurrences of a substring within a string.  <b>Example:</b> <pre>&lt;cfset myString = "ColdFusion"&gt; &lt;cfoutput&gt;#Replace(myString, "Cold", "Hot", "ALL")# &lt;/cfoutput&gt; &lt;!-- Output: HotFusion --&gt;</pre>
---	--

## Date and Time Functions

<code>Now()</code>	Returns the current date and time.  <b>Example:</b> <pre>&lt;cfoutput&gt;#Now()# &lt;/cfoutput&gt;</pre>
<code>DateFormat(date, mask)</code>	Formats a date according to a specified mask.  <b>Example:</b> <pre>&lt;cfset myDate = Now()&gt; &lt;cfoutput&gt;#DateFormat(myDate, "mm/dd/yyyy")# &lt;/cfoutput&gt;</pre>
<code>TimeFormat(time, mask)</code>	Formats a time according to a specified mask.  <b>Example:</b> <pre>&lt;cfset myTime = Now()&gt; &lt;cfoutput&gt;#TimeFormat(myTime, "HH:mm:ss")#&lt;/cfoutput&gt;</pre>
<code>DateAdd(interval, number, date)</code>	Adds a specified interval to a date.  <b>Example:</b> <pre>&lt;cfset myDate = Now()&gt; &lt;cfoutput&gt;#DateAdd("d", 7, myDate)#&lt;/cfoutput&gt;</pre>
<code>DateDiff(interval, date1, date2)</code>	Calculates the difference between two dates.  <b>Example:</b> <pre>&lt;cfset date1 = CreateDate(2023, 1, 1)&gt; &lt;cfset date2 = Now()&gt; &lt;cfoutput&gt;#DateDiff("d", date1, date2)#&lt;/cfoutput&gt;</pre>
<code>CreateDate(year, month, day)</code>	Creates a date object.  <b>Example:</b> <pre>&lt;cfset myDate = CreateDate(2023, 10, 26)&gt; &lt;cfoutput&gt;#myDate# &lt;/cfoutput&gt;</pre>

## Array Functions

<code>ArrayNew(dimensions)</code>	Creates a new array.  <b>Example:</b> <pre>&lt;cfset myArray = ArrayNew(1)&gt;</pre>
<code>ArrayAppend(array, value)</code>	Appends an element to an array.  <b>Example:</b> <pre>&lt;cfset myArray = ArrayNew(1)&gt; &lt;cfset ArrayAppend(myArray, "Value1")&gt;</pre>
<code>ArrayLen(array)</code>	Returns the length of an array.  <b>Example:</b> <pre>&lt;cfset myArray = ["A", "B", "C"]&gt; &lt;cfoutput&gt;#ArrayLen(myArray) #&lt;/cfoutput&gt;</pre>
<code>ArrayAvg(array)</code>	Returns the average of the elements in an array.  <b>Example:</b> <pre>&lt;cfset myArray = [1, 2, 3, 4, 5]&gt; &lt;cfoutput&gt;#ArrayAvg(myArray) #&lt;/cfoutput&gt;</pre>
<code>ArrayMin(array)</code>	Returns the minimum value in an array.  <b>Example:</b> <pre>&lt;cfset myArray = [5, 2, 8, 1, 9]&gt; &lt;cfoutput&gt;#ArrayMin(myArray) #&lt;/cfoutput&gt;</pre>
<code>ArrayMax(array)</code>	Returns the maximum value in an array.  <b>Example:</b> <pre>&lt;cfset myArray = [5, 2, 8, 1, 9]&gt; &lt;cfoutput&gt;#ArrayMax(myArray) #&lt;/cfoutput&gt;</pre>

# Components and Objects

## Component Definition

<code>&lt;cfcomponent</code>	Defines a CFML component (CFC).
<code>&gt;</code>	<b>Example:</b> <pre>&lt;cfcomponent name="MyComponent"&gt;   &lt;cffunction name="init"&gt;     &lt;cfreturn this&gt;   &lt;/cffunction&gt; &lt;/cfcomponent&gt;</pre>
<code>&lt;cffunction</code>	Defines a function within a component.
<code>&gt;</code>	<b>Example:</b> <pre>&lt;cffunction name="myFunction" access="public" returnType="string"&gt;   &lt;cfargument name="param1" type="string" required="true"&gt;   &lt;cfreturn "Hello, #arguments.param1#"&gt; &lt;/cffunction&gt;</pre>
<code>access</code>	Specifies the access level of a function (public, private, remote).
	<b>Example:</b> <pre>&lt;cffunction name="privateFunc" access="private"&gt;   &lt;!--- Only accessible within the component ---&gt; &lt;/cffunction&gt;</pre>
<code>returnType</code>	Specifies the return type of a function.
	<b>Example:</b> <pre>&lt;cffunction name="getDate" returnType="date"&gt;   &lt;cfreturn Now()&gt; &lt;/cffunction&gt;</pre>
<code>&lt;cfargument</code>	Defines an argument for a function.
<code>&gt;</code>	<b>Example:</b> <pre>&lt;cffunction name="greet"&gt;   &lt;cfargument name="name" type="string" required="true"&gt;   &lt;cfreturn "Hello, #arguments.name#"&gt; &lt;/cffunction&gt;</pre>

## Object Creation and Usage

<code>CreateObject(type,</code> <code>componentPath)</code>	Creates an instance of a CFML component.
	<b>Example:</b> <pre>&lt;cfset myObj = CreateObject("component", "path.to.MyComponent")&gt; &lt;cfoutput&gt;#myObj.myFunction("World")# &lt;/cfoutput&gt;</pre>
Using dot notation	Accessing properties and methods of an object.
	<b>Example:</b> <pre>&lt;cfset obj = CreateObject("component", "MyComponent")&gt; &lt;cfoutput&gt;#obj.propertyName#&lt;/cfoutput&gt; &lt;cfoutput&gt;#obj.methodName()#&lt;/cfoutput&gt;</pre>