



[www.coldfusionbrasil.com.br](http://www.coldfusionbrasil.com.br)

© 2000

<b>USANDO CFSWITCH, CFCASE E CFDEFAULTCASE</b>	<b>3</b>
UTILIZANDO CFSWITCH, CFCASE E CFDEFAULTCASE COM UMA VARIÁVEL	4
UTILIZANDO CFSWITCH, CFCASE E CFDEFAULTCASE COM UMA VARIÁVEL E O PARÂMETRO DELIMITERS.	4
UTILIZANDO CFSWITCH, CFCASE E CFDEFAULTCASE COM UMA QUERY.	5
<b>FUNÇÕES</b>	<b>5</b>
<b>FUNÇÕES DE DATA</b>	<b>6</b>
- DAY	6
- MONTH	6
- YEAR	6
<b>FUNÇÕES DE HORA</b>	<b>7</b>
- HOUR	7
- MINUTE	7
- SECOND	7
- NOW	7
<b>UTILIZANDO AS FUNÇÕES DE DATA E HORA</b>	<b>8</b>
<b>FUNÇÕES DE FORMATAÇÃO</b>	<b>8</b>
- DATEFORMAT	8
- TIMEFORMAT	9
<b>UTILIZANDO AS FUNÇÕES DE FORMATAÇÃO DATEFORMAT E TIMEFORMAT</b>	<b>9</b>
- URLENCODEDFORMAT	10
<b>FUNÇÕES MATEMÁTICAS</b>	<b>10</b>
- MAX	10
- MIN	10
- RANDRANGE	11
<b>FUNÇÕES DE STRING</b>	<b>11</b>
- LEN	11
- LTRIM	11
- RTRIM	11

## Usando CFSWITCH, CFCASE e CFDEFAULTCASE

Começaremos este tutorial vendo como tratar condições de múltiplas opções com as tags <CFSWITCH>, <CFCASE> e <CFDEFAULTCASE>. Estas três tags são usadas em conjunto sendo a CFSWITCH responsável pela expressão, a CFCASE pelas comparações e CFDEFAULTCASE para quando nenhum valor atende as comparações feitas pela CFCASE.

Veremos agora a estrutura de utilização dessas três tags:

```
<CFSWITCH EXPRESSION="expressão">
    <CFCASE VALUE="valor de comparação" DELIMITERS="delimitadores">
        tags CFML ou HTML
    </CFCASE>
    comparações adicionais <CFCASE></CFCASE>
    <CFDEFAULTCASE>
        tags CFML ou HTML
    </CFDEFAULTCASE>
</CFSWITCH>
```

Abaixo entenderemos cada parâmetro passado nestas tags:

Parâmetro	Descrição
EXPRESSION	Obrigatório. Aqui deve ser colocado a expressão/variável que irá ter seu valor comparado.
VALUE	Obrigatório. Este parâmetro é responsável pelos valores que serão comparados a expressão.
DELIMITERS	Opcional. Define que caracter separa uma lista de valores no parâmetro VALUE. Seu valor padrão é a vírgula (,).

Como podemos ver, utilizando estas tags poderemos substituir uma série de CFIFs aninhados. Para melhor entendermos teremos abaixo três exemplos de como utilizar-los.

### Utilizando CFSWITCH, CFCASE e CFDEFAULTCASE com uma variável

```
<CFSET valor1=1>
<CFSET valor2=2>

<CFSET total=#valor1#+#valor2#>
<CFSWITCH EXPRESSION="#total#">
    <CFCASE VALUE="1">
        Um
    </CFCASE>
    <CFCASE VALUE="2">
        Dois
    </CFCASE>
    <CFCASE VALUE="3">
        Três
    </CFCASE>
    <CFDEFAULTCASE>
        Nenhuma das anteriores
    </CFDEFAULTCASE>
</CFSWITCH>
```

No exemplo acima como a variável TOTAL é a soma das outras duas variáveis (VALOR1, VALOR2), teremos como resposta “Três”.

### Utilizando CFSWITCH, CFCASE e CFDEFAULTCASE com uma variável e o parâmetro DELIMITERS.

```
<CFSET valor1=1>
<CFSET valor2=2>

<CFSET total=#valor1#+#valor2#>
<CFSWITCH EXPRESSION="#total#">
    <CFCASE VALUE="1;4" DELIMITERS=";">
        Um
    </CFCASE>
    <CFCASE VALUE="2">
        Dois
    </CFCASE>
    <CFCASE VALUE="3">
        Três
    </CFCASE>
    <CFDEFAULTCASE>
        Nenhuma das anteriores
    </CFDEFAULTCASE>
</CFSWITCH>
```

Note que o exemplo acima só houve mudanças no primeiro CFCASE, que retornará “Um” caso a variável TOTAL seja igual a 1 ou 4.

### Utilizando CFSWITCH, CFCASE e CFDEFAULTCASE com uma query.

```
<CFQUERY DATASOURCE="coldfusionbrasil" NAME="Pesquisa">
    select * from cadastro
</CFQUERY>

<CFOUTPUT QUERY="Pesquisa">
<CFSWITCH EXPRESSION="#cidade#">
    <CFCASE VALUE="São Paulo">
        Moradores de São Paulo:<BR>
        #nome# - #email#
    </CFCASE>
    <CFCASE VALUE="Rio de Janeiro">
        Moradores do Rio de Janeiro:<BR>
        #nome# - #email#
    </CFCASE>
    <CFCASE VALUE="Santos">
        Moradores de Santos:<BR>
        #nome# - #email#
    </CFCASE>
    <CFDEFAULTCASE>
        Moradores de outras localidades:<BR>
        #nome# - #email#
    </CFDEFAULTCASE>
</CFSWITCH>
</CFOUTPUT>
```

Vimos neste exemplo que podemos também trabalhar com resultados de uma query. No exemplo acima os nomes são classificados de acordo com sua localidade.

---

## Funções

Estaremos vendo agora algumas funções muito importantes para o desenvolvimento de aplicações em ColdFusion. Irão encontrar aqui funções que tratam data, hora, formatação de dados, matemáticas e strings.

## Funções de data

### - **Day**

Retorna o dia, de 1 a 31, para uma data.

Sintaxe: Day(data)

Exemplo:

```
<CFSET dia_venda = Day("01/02/2000")>
```

Note que o valor da variável DIA\_VENDA será 02 e não 01, pois a função utiliza a formatação mês/dia/ano.

### - **Month**

Retorna o mês, de 1 a 12, para a data.

Sintaxe: Month(data)

Exemplo:

```
<CFSET mes_venda = Month("01/02/2000")>
```

A exemplo da função Day, o mesmo acontece aqui; o resultado da variável MES\_VENDA é 01 e não 02.

### - **Year**

Retorna o ano de uma data.

Sintaxe: Year(data)

Exemplo:

```
<CFSET ano_venda = Year("01/02/2000")>
```

## Funções de Hora

### - *Hour*

Retorna a hora, entre 0 e 23, de um horário.

Sintaxe: Hour(horário)

Exemplo:

```
<CFSET hora_venda = Hour ("15:07:33")>
```

### - *Minute*

Retorna os minutos de um horário.

Sintaxe: Minute(horário)

Exemplo:

```
<CFSET minuto_venda = Minute("15:07:33")>
```

### - *Second*

Retorna os segundos de um horário.

Sintaxe: Second(horário)

Exemplo:

```
<CFSET segundo_venda = Second("15:07:33")>
```

Agora veremos uma função que é muito útil tanto para a utilização com funções de data quanto para funções de hora.

### - *Now*

Esta função retorna a data e hora do sistema, em seu servidor Cold Fusion.

Sintaxe: Now()

Exemplo:

```
<CFSET data_atual = Now()>
```

## Utilizando as funções de Data e Hora

Para termos uma maior compreensão da utilização destas funções iremos ver agora como utiliza-las na prática, através de um exemplo que incorpora todas as funções passadas até agora.

```
<CFSET data_atual = Now()>
```

```
<CFSET dia_atual = Day(Now())>
```

```
<CFSET mes_atual = Month(Now())>
```

```
<CFSET ano_atual = Year(Now())>
```

```
<CFSET hora_atual = Hour(Now())>
```

```
<CFSET minuto_atual = Minute(Now())>
```

```
<CFSET segundo_atual = Second(Now())>
```

Criando as variáveis com as funções de data e hora.

```
<CFOUTPUT>
```

```
Data Gerada pelo sistema: #data_atual#<p>
```

```
Dia: #dia_atual# Mês: #mes_atual# Ano: #ano_atual#<br>
```

```
Data Gerada pela aplicação: #dia_atual#/#mes_atual#/#ano_atual#<p>
```

```
Hora: #hora_atual# Minuto: #minuto_atual# Segundo: #segundo_atual#<br>
```

```
Horário Gerado pela aplicação: #hora_atual#:#minuto_atual#:#segundo_atual#
```

```
</CFOUTPUT>
```

## Funções de Formatação

### - *DateFormat*

Retorna uma data formatada de acordo com uma máscara especificada. Se nenhuma máscara for definida, a função retorna a data usando o formato dd-mm-yy.

Sintaxe: `DateFormat(data, máscara)`

Máscara:

- d – dia sem zero a esquerda
- dd – dia com dois dígitos incluindo o zero
- m – mês sem zero a esquerda
- mm – mês com dois dígitos incluindo o zero
- mmm – abreviação do mês com as três primeiras letras
- mmmm – mês por extenso
- yy – ano com dois dígitos
- yyyy – ano com quatro dígitos



Exemplo:

```
<CFSET data_venda = ("01/02/2000","dd/mm/yyyy")>
```

Teremos como resultado: 02/01/2000, lembrando que o formato padrão do ColdFusion é mês/dia/ano.

### **- TimeFormat**

Esta função retorna o horário formatado de acordo com uma máscara especificada. Se nenhuma máscara for definida, por padrão será aceito o formato hh:mm tt.

Sintaxe: TimeFormat(horário, máscara)

Máscara:

- h – hora sem zero a esquerda. Relógio de 12h
- hh – hora com dois dígitos incluindo o zero. Relógio de 12h
- H – hora sem zero a esquerda. Relógio de 24h
- HH – h ora com dois dígitos incluindo o zero. Relógio de 24h
- m – minutos sem zero a esquerda
- mm – minutos com dois dígitos incluindo o zero
- s – segundos sem zero a esquerda
- ss – segundos com dois dígitos incluindo o zero
- t – identifica o horário como sendo A (antes meio-dia) ou P (pós meio-dia)
- tt – identifica o horário como sendo AM ou PM

Exemplo:

```
<CFSET hora_atual = TimeFormat("13:54:22","hh:mm:ss tt")>
```

Teremos como resultado: 01:54:22PM.

### **Utilizando as funções de formatação DateFormat e TimeFormat**

Vimos no exemplo anterior como utilizar na prática as funções de data e hora, e agora poderemos otimizar este exemplo utilizando as funções DateFormat e TimeFormat. Vejamos como:

```
<CFSET data_atual = DateFormat(Now(),"dd/mm/yyyy")>
```

```
<CFSET hora_atual = TimeFormat(Now(),"HH:mm:ss")>
```

```
<CFOUTPUT>
```

```
Data Gerada pelo sistema: #Now()#<p>
```

```
Data Gerada pela aplicação: #data_atual#<p>
```

Horário Gerado pela aplicação: #hora\_atual#  
</CFOUTPUT>

### **- URLEncodedFormat**

Esta função converte uma string para formato de URL. Ou seja, se você tiver uma string com espaços, esta função substitui estes espaços por caracteres especiais, para que a string possa ser passada através da URL.

Sintaxe: URLEncodedFormat(string)

Exemplo:

```
<CFSET nome = "José Joaquim Silva">
```

```
<CFSET nomeURL = URLEncodedFormat("nome")>
```

Como resultado teremos: Jos%E9%20Joaquim%20Silva

## **Funções Matemáticas**

### **- Max**

Retorna o valor maior entre dois números.

Sintaxe: Max(número1, número2)

Exemplo:

```
<CFSET maior = Max(1,2)>
```

### **- Min**

Retorna o valor menor entre dois números.

Sintaxe: Min(número1,número2)

Exemplo:

```
<CFSET menor = Min(1,2)>
```

### **- RandRange**

Retorna um número aleatório entre dois números. Os números devem ser menores que 100.000.000.

Sintaxe: RandRange(número1, número2)

Exemplo:

```
<CFSET sorteio = (1,35)>
```

### **Funções de String**

#### **- Len**

Esta função retorna o total de caracteres de uma string.

Sintaxe: Len(string)

Exemplo:

```
<CFSET tamanho = Len("coldfusionbrasil")>
```

#### **- LTrim**

Remove todos os espaços em branco à esquerda de uma string.

Sintaxe: LTrim(string)

Exemplo:

```
<CFSET palavra = LTrim(" coldfusionbrasil")>
```

#### **- RTrim**

Remove todos os espaços em branco à direita de uma string.

Sintaxe: RTrim(string)

Exemplo:

```
<CFSET palavra = RTrim("coldfusionbrasil ")>
```