



www.coldfusionbrasil.com.br

© 2000

PASSANDO VARIÁVEIS ATRAVÉS DE FORM E URL	3
ALTERA.CFM	3
FORMALTERA.CFM	4
ALTERA.CFM	4
FORMALTERA.CFM	4
CRIANDO E MANIPULANDO VARIÁVEIS COM CFSET	4
VALORES ESTÁTICOS	5
PARÂMETROS DINÂMICOS	5
EXPRESSÕES	6
CRIANDO COOKIES	6
VERIFICANDO A EXISTÊNCIA DE UMA VARIÁVEL	7
DEFININDO UM VALOR PADRÃO PARA VARIÁVEIS	7
USO PRÁTICO DA FUNÇÃO ISDEFINED E A TAG CFPARAM	7
PARAM1.CFM	8
PARAM2.CFM	8
MOSTRAR AS VARIÁVEIS	8

Passando Variáveis através de Form e URL

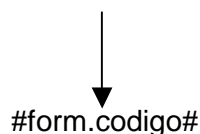
No último módulo vimos como inserir, atualizar e excluir registros em um banco de dados, e em dado momento, foi mostrado uma instrução da seguinte forma:

#form.codigo#

“Mas o que significa isto?”, deve você ter se perguntado. Esta instrução significa uma variável que foi passada através de um formulário para uma aplicação cfm. Esta aplicação irá interpretar o seu conteúdo e executar a tarefa referente a ela.

Portanto, a instrução acima está ligada a uma sub-tag de um formulário:

```
<INPUT TYPE="HIDDEN" NAME="codigo" VALUE="codigo">
```



Um outro modo de passar variáveis/atributos para uma aplicação cfm é através de URL. Você já deve ter visto em alguns sites algo semelhante a isto:

<http://dominio/cgi-bin/aplicacao.cgi?variavel=valor>

Neste caso a aplicação recebe a variável, que vem identificada logo após o ponto de interrogação (?) junto com seu valor, e não mais através de um formulário como mostrado anteriormente.

Na aplicação esta variável será referenciada de forma semelhante ao de variáveis passadas através de um formulário. Basta substituir form por url:

#url.variável#

Para passar mais de uma variável para a aplicação, basta separar o conjunto variável e valor por um &.

<http://dominio/cgi-bin/aplicacao.cgi?variavel1=valor1&variavel2=valor2>

Para entendermos melhor a utilização destas duas instruções, vamos rever a aplicação utilizada no último módulo, e vamos montar esta mesma aplicação passando então as variáveis necessárias através de URL.

Altera.cfm

```
<CFQUERY DATASOURCE="agenda" NAME="AgendaPessoal">
  SELECT * FROM Pessoal
</CFQUERY>

<CFOUTPUT QUERY="AgendaPessoal">
  #nome# - #tel# -
  <FORM METHOD="POST" ACTION="formaltera.cfm">
    <INPUT TYPE="HIDDEN" NAME="codigo" VALUE="#codigo#">
    <INPUT TYPE="SUBMIT" VALUE="Alterar">
  </FORM>
</CFOUTPUT>
```

Variável que está sendo passada para a próxima aplicação.

FormAlterar.cfm

```
<CFQUERY DATASOURCE="agenda" NAME="AgendaPessoal">
  SELECT * FROM Pessoal
  WHERE codigo = #form.codigo#
</CFQUERY>

<CFOUTPUT QUERY="AgendaPessoal">
  <FORM METHOD="POST" ACTION="AlterarFim.cfm">
    <INPUT TYPE="TEXT" NAME="nome" VALUE="#nome#">
    <INPUT TYPE="TEXT" NAME="tel" VALUE="#tel#">
    <INPUT TYPE="HIDDEN" NAME="codigo" VALUE="#codigo#">
    <INPUT TYPE="SUBMIT" VALUE="Alterar">
  </FORM>
</CFOUTPUT>
```

Variável passada pelo formulário

Agora vejamos como fazer isto com uma URL.

Alterar.cfm

```
<CFQUERY DATASOURCE="agenda" NAME="AgendaPessoal">
  SELECT * FROM Pessoal
</CFQUERY>

<CFOUTPUT QUERY="AgendaPessoal">
  #nome# - #tel# -
  <a href="formalterar.cfm?codigo=#codigo#">Alterar</a>
</CFOUTPUT>
```

Variável sendo passada através de URL

FormAlterar.cfm

```
<CFQUERY DATASOURCE="agenda" NAME="AgendaPessoal">
  SELECT * FROM Pessoal
  WHERE codigo = #url.codigo#
</CFQUERY>

<CFOUTPUT QUERY="AgendaPessoal">
  <FORM METHOD="POST" ACTION="AlterarFim.cfm">
    <INPUT TYPE="TEXT" NAME="nome" VALUE="#nome#">
    <INPUT TYPE="TEXT" NAME="tel" VALUE="#tel#">
    <INPUT TYPE="HIDDEN" NAME="codigo" VALUE="#codigo#">
    <INPUT TYPE="SUBMIT" VALUE="Alterar">
  </FORM>
</CFOUTPUT>
```

Variável passada pela URL

Criando e Manipulando Variáveis com CFSET

Vimos como são passadas variáveis através de aplicações, agora vamos ver como criar variáveis na mesma aplicação. A tag CFSET é a responsável pela criação de variáveis e a manipulação de seus valores.

Com esta tag, você poderá usar valores estáticos, parâmetros dinâmicos ou expressões para criar uma variável. A seguir é mostrada a sua sintaxe:

`<CFSET Nome_da_Variavel = Valor Estático, Parâmetro ou Expressão>`

Vejamos agora alguns exemplos da utilização destes tipos:

Valores Estáticos

Vamos criar uma variável chamada *verifica* e colocar seu valor como *José da Silva*:

`<CFSET verifica = 'José da Silva'>`

Portanto em nossa aplicação *agenda* ilustrada no primeiro módulo deste tutorial poderíamos estar utilizando algo assim:

```
<CFOUTPUT QUERY="AgendaPessoal">
  #nome# - #tel#
  <CFSET verifica = 'José da Silva'>
  <CFIF nome IS verifica>
    <br>Zé
  <CFELSE>
    <br>Outros
  </CFIF>
  <HR>
</CFOUTPUT>
```

Agora para criar uma variável com um valor numérico estático, basta usarmos:

`<CFSET nome_variavel = 54>`

Parâmetros Dinâmicos

A tag CFSET também aceita parâmetros passados por outras aplicações, como variáveis passadas através de formulários e URL.

Para criarmos uma variável que tenha como valor o conteúdo de uma variável passada por um formulário, basta usarmos:

`<CFSET nome_variavel = form.variavel>`

Um outro tipo aceito pela CFSET é a combinação de parâmetro dinâmico com string:

`<CFSET nome_variavel = '#PARÂMETRO# texto'>`

Em nossa aplicação *agenda* poderíamos ter algo assim:

```
<CFOUTPUT QUERY="AgendaPessoal">
  #nome# - #tel#
<CFSET verifica = 'José da Silva'>
<CFIF nome IS verifica>
  <CFSET msg = '#nome# seu apelido é Zé'>
<CFELSE>
  <CFSET msg = '#nome# você não tem apelido'>
</CFIF>
<HR>
</CFOUTPUT>
```

Expressões

A tag CFSET também aceita como seu conteúdo expressões matemáticas e de concatenação de strings e valores numéricos.

Em,

```
<CFSET ValorTotal = 2 * (10 + 5)>
```

Criamos uma expressão matemática;

Em,

```
<CFSET Divida = 'José da Silva você deve ' & ValorTotal>
```

Criamos uma concatenação de string com valor numérico.

Criando Cookies

Cookies são mecanismos utilizados para armazenar informações de cada usuário que visita seu site. O Cold Fusion pode utilizar estas informações, armazenadas nos browsers, em suas aplicações. A tag que o Cold Fusion usa para isto é o CFCOOKIE.

A tag CFCOOKIE tem os seguintes atributos:

NAME	Requerido. Nome da variável cookie a ser criada.
VALUE	Opcional. Indica o valor determinado para a variável
EXPIRES	Opcional. Este atributo determina o tempo de expiração do cookie. Pode ser especificado uma data (como, 06/06/1999), número de dias (como, 1, 10, 100), NOW, ou NEVER. Utilizando NOW o cookie é apagado do browser do usuário.
SECURE	Opcional. Indica que a variável deve ser transmitida seguramente. Se o browser não suportar SSL, o cookie não é enviado.
PATH	Opcional. Especifica os subdiretórios de URLs em que o cookie é aplicado.
DOMAIN	Opcional. Especifica o domínio em que o cookie é valido.

Vejamos agora como criar um cookie:

```
<CFCOOKIE NAME="ID" VALUE="1234" EXPIRES="1">
```

Acima criamos um cookie com o nome *ID*, de valor *1234* e que irá expirar em *1* dia.

Verificando a existência de uma variável

Em muitos casos, algumas aplicações necessitam que sejam passadas determinadas variáveis, e caso estas não sejam passadas é apresentado uma mensagem de erro padrão do Cold Fusion. Para que isto não ocorra, podemos utilizar uma função do Cold Fusion que verifica a existência de variáveis.

A função utilizada para isto é *ISDEFINED*. Abaixo segue sua sintaxe:

```
IsDefined ("nome da variável")
```

Esta função verifica se a variável especificada existe, se sim retorna *TRUE* e caso contrário retorna *FALSE*.

Definindo um valor padrão para variáveis

Definir valores padrões de variáveis em algumas aplicações pode ser muito útil, para se fazer isto, basta utilizarmos a tag *CFPARAM*.

A tag *CFPARAM* pode ser usada para variáveis criadas pela tag *CFSET*, formulários, URL e Cookie. Sua sintaxe é a seguinte:

```
<CFPARAM NAME="nome da variável" DEFAULT="valor">
```

Onde:

NAME	É o nome da variável
DEFAULT	O valor padrão da variável

Uso prático da função *IsDefined* e a tag *CFPARAM*

Já sabendo como funcionam a função *IsDefined* e a tag *CFPARAM*, vamos ver agora como utilizar ambos em uma única aplicação e entender melhor onde aplicá-las.

Vamos criar uma aplicação que terá dois links para uma mesma aplicação. Sua única diferença será que em um deles iremos colocar uma variável chamada *teste* com seu respectivo valor.

Nosso primeiro arquivo *cfm* ficará assim:

Param1.cfm

```
<a href=param2.cfm?teste="Link-1">Link-1</a><br>
<a href=param2.cfm>Link-2</a>
```

O segundo arquivo:

Param2.cfm

```
<CFIF IsDefined("url.teste")>
  <CFOUTPUT>#url.teste#</CFOUTPUT>
<CFELSE>
  <CFPARAM NAME="url.teste" DEFAULT="Link-2">
  <CFOUTPUT>#url.teste#</CFOUTPUT>
</CFIF>
```

Teremos como resultado caso seja clicado o primeiro link, a mensagem *Link 1*, e se for o segundo link a ser clicado, a mensagem será *Link 2*.

Mostrar as variáveis

Até agora vimos e entendemos como criar e verificar a existência de uma variável. Mas como fazer para mostrar seu conteúdo ?

No último exemplo foi dada uma pequena amostra de como fazer isto:

```
<CFOUTPUT>#url.teste#</CFOUTPUT>
```

Para mostrar os vários tipos de variáveis aceitos pelo Cold Fusion, basta utilizar alguns dos conceitos já aprendidos neste tutorial, a tag CFOUTPUT e o sinal de #, mais um identificador do tipo de variável. Portanto teremos:

```
<CFOUTPUT>#tipo_da_variável.variável#</CFOUTPUT>
```

Abaixo temos um quadro que ilustra como proceder para cada tipo de variável:

Variável criada por:	Sintaxe
Form	<CFOUTPUT>#form.variável#</CFOUTPUT>
URL	<CFOUTPUT>#url.variável#</CFOUTPUT>
Cookie	<CFOUTPUT>#cookie.variável#</CFOUTPUT>
CFSET	<CFOUTPUT>#variável#</CFOUTPUT>

* Note que as variáveis criadas pela tag CFSET não precisam de identificador de tipo.