



ASP

Autor:
Nicleverson Silva

ASP

Active Server Pages

Curso básico

CURSO MASTER

Página 1

INTRODUÇÃO

A Internet é um conjunto de redes de computadores interligados pelo mundo inteiro, onde um misto de protocolos e serviços permitem ao usuário obter acesso à informação e se comunicar remotamente.

Trata-se da mais bem sucedida aplicação prática do conceito de interoperabilidade, conectividade de redes de tecnologias distintas.

A criação da World Wide Web (WWW) popularizou a rede. Este serviço para a transmissão multimídia de informações, implementado pelo protocolo de aplicação HTTP (Hypertext Transfer Protocol),

realiza uma comunicação cliente-servidor, com base na requisição, visualização e transferência de arquivos.

Geralmente, os arquivos se apresentam no formato HTML (Hypertext Markup Language), podendo conter referências a outros tipos de arquivos (imagens, sons, vídeos,...).

Esse modelo de funcionamento limitou bastante o uso da Web, uma vez que as páginas HTML têm um conteúdo estático, ou seja, sempre são exibidas da mesma forma e não possibilitam nenhuma interação com o usuário.

Para deixar a Web mais dinâmica e interativa, criou-se o CGI (Common Gateway Interface). O programa é processado e o resultado desse processamento é enviado pelo servidor Web ao cliente,

geralmente no formato HTML. O dinamismo do CGI está no fato do processamento poder retornar diferentes resultados, dependendo dos parâmetros informados pelo cliente (interação) ao programa CGI.

Apesar de dar mais “vida” a Web, programas CGI possuem uma série de desvantagens técnicas, sendo a principal delas o fato de tais programas serem executados de forma diferenciada no Web Server.

Um servidor Web que recebe várias requisições simultâneas, facilmente se sobrecarrega e pára.

Por este motivo, surgiram, e ainda surgem a cada dia, tecnologias alternativas ao uso do CGI:

ISAPI, NISAPI, IDC/HTX, Cold Fusion, Java Server Pages (JSP), Personal Home Page (PHP), Active

Server Pages (ASP),...

ACTIVE SERVER PAGES

ASP é uma tecnologia desenvolvida pela Microsoft que disponibiliza um conjunto de componentes para o desenvolvimento de páginas Web dinâmicas. Tais páginas consistem em arquivos de extensão .asp

no formato texto (ASCII) que contêm combinações de scripts e tags HTML.

Um servidor Web que suporta ASP funciona da seguinte forma:

- Cliente solicita página *.asp;

- Servidor abre a página e lê seu conteúdo:

pagina 2

Se encontra tags HTML, envia direto ao cliente;

Se encontra comandos de script:

- Pára o envio;
- Processa os comandos;
- Envia o resultado HTML ao cliente.

Como todo código de programação existente em páginas ASP é executado no servidor, e este só retorna ao cliente respostas em HTML, aplicações ASP têm seu código fonte totalmente preservado, além

de poderem ser acessadas por qualquer tipo de browser existente no mercado.

Entre os recursos que podem ser implementados com ASP, podemos citar:

- Programação com Visual Basic Script e Java Script;
- Acesso a banco de dados;
- Envio de e-mail.

Para utilizar ASP, o computador que a hospeda as páginas (Web Server) deve possuir como sistema operacional o Windows NT Server 4.0 (ou superior) com o Internet Information Server 3.0 (ou

superior) O IIS é um programa servidor Web da Microsoft. Se as páginas utilizarem recursos de acesso a

banco de dados, será necessário um driver de ODBC instalado e funcionando no servidor.

O ASP também “funciona” com o MS Personal Web Server (PWS), para Windows NT WorkStation e para Windows 9x, muito embora essa não seja a plataforma mais recomendada. Para as plataformas

Unix / Linux, já existem módulos no mercado que garantem o suporte a ASP.

ROTINAS DE SCRIPT

Script é um programa escrito numa determinada linguagem de programação que não necessita ser compilado para ser posteriormente executado. Scripts são interpretados, ou seja, seus comandos são lidos

em tempo de execução por um Script Engine, processados e seus resultados passados para a saída

padrão da aplicação (monitor de vídeo, impressora, servidor Web,...).

Toda a funcionalidade de uma página ASP é controlada através de comandos de Script.

Teoricamente, o ASP pode utilizar qualquer Script Engine (interpretador), mas na prática, a Microsoft só

disponibiliza dois:

- Visual Basic Script (VBScript) - default
- MS Java Script(JScript)

Ao escrevemos páginas *.ASP, a primeira coisa que devemos fazer é indicar em qual dessas linguagens disponíveis elas serão escritas:

<% @ LANGUAGE = VBScript %>

ou

<SCRIPT LANGUAGE = “VBScript” RUNAT = SERVER> ... </SCRIPT>

Como um Web browser reconhece um script? Ao encontrar na página *.ASP a tag <% ou

<SCRIPT LANGUAGE = “VBScript” RUNAT = SERVER>, o browser entende que daquela posição até

%> ou **</SCRIPT>** existem comandos de script a serem executados. É importante observar que o Web

Server só tentará interpretar uma página se a mesma estiver salva com a extensão .asp, caso contrário, o

servidor Web enviará a página como se fosse um arquivo de texto normal. Logo, não adianta criar scripts

altamente eficientes e esquecer de salvar corretamente o arquivo.

Página 3

Exemplo 1:

Página - **now.asp** - que retorna a data e a hora atual do servidor Web:

```
<% @ LANGUAGE=VBSCRIPT %>
<HTML><HEAD><TITLE> ASP - curso básico </TITLE></HEAD>
<BODY>
<%=NOW%> <!--função do vbscript que retorna hora/data-->
</BODY></HTML>
```

VISUAL BASIC SCRIPT (VBSCRIPT)

VBScript é uma linguagem criada a partir do Visual Basic, mas com algumas limitações por motivos de segurança, além de ser interpretada e não compilada. Permite a manipulação de strings, datas,

numéricos e objetos Active X do servidor.

Sendo a linguagem de script mais utilizada para desenvolvimento de páginas ASP, é de fundamental importância conhecer seus principais comandos.

Quando a página ASP encontra um erro, o mesmo é bem ilustrado no browser, facilitando bastante o trabalho de depuração para os programadores ASP.

Regra 1: VBScript só aceita um comando por linha:

correção: separar os comandos por : (dois pontos).

Exemplo 2: linha1.asp:

```
<% @ LANGUAGE=VBSCRIPT %>
<HTML><HEAD><TITLE> ASP - curso básico </TITLE></HEAD>
<BODY>
<% a = 2 b = a*2 %> <!--erro-->
<% a = 2 : b = a*2 %> <!--correção-->
Valor de B = <%=b%>
</BODY></HTML>
```

Regra2: um comando não pode existir em mais de uma linha:

correção: colocar o comando numa única linha ou usar o caractere _ (underline).

Exemplo 3: linha2.asp:

```
<% @ LANGUAGE=VBSCRIPT %>
<HTML><HEAD><TITLE> ASP - curso básico </TITLE></HEAD>
<BODY>
<% texto = "Paulo Henrique" +
"Guaraciaba"%> <!--erro-->
<% texto = "Paulo Henrique" + _
"Guaraciaba"%> <!--correção-->
<% texto = "Paulo Henrique" + "Guaraciaba"%> <!--correção-->
Nome = <%=texto%>
</BODY></HTML>
```

VARIÁVEIS

São identificadores alfanuméricos que "apontam" para posições de memória onde existem valores

Pagina4

armazenados temporariamente, sendo que estes podem ser alterados durante o processamento de uma

aplicação. Não nos interessa saber como esse valor será armazenado na memória, nem onde.

Basta

apenas sabermos o nome e o tipo do valor armazenado em tal variável.

Em VBScript os nomes de variáveis devem começar obrigatoriamente com uma letra e não podem exceder 255 caracteres. Ao contrário da maioria das linguagens de programação, uma variável do VBScript não necessita ser declarada antes de ser utilizada. Entretanto, scripts com declarações implícitas

de variáveis são mais difíceis de ser entendidos, além de estarem mais vulneráveis a erros de digitação.

Exemplo 4: var.asp:

```
<% @ LANGUAGE=VBSCRIPT %>
<% Option Explicit %> <!--torna obrigatória a declaração-->
<HTML><HEAD><TITLE>Curso ASP</TITLE></HEAD>
<BODY>
<% Dim v1 <!--declaração explícita-->
v1 = 100
v2 = 200
v3 = 300 %>
V1=<%=v1%><BR>
V2=<%=v2%><BR>
V3=<%=v3%>
</BODY></HTML>
```

Obs.: O “tempo de vida” de uma variável vai desde sua declaração explícita (Dim) ou implícita (sem Dim) até o final do script ou sub-rotina.

TIPOS DE DADOS

O VBScript contém apenas um tipo de variável chamado de **variant**, que pode armazenar valores de qualquer tipo. Só que, num determinado instante, esta variável possui apenas um tipo implícito, determinado pelo valor a ela atribuído.

Subtipos:

- Integer
- Long
- Single
- Double
- Date
- String
- Boolean
- Null
- Empty
- Object
- Array (T)

Obs.: Arrays têm que ser declarados explicitamente, mas cada elemento de um array pode ser de um subtipo diferente.

Exemplo 5: tipos.asp:

```
<% @ LANGUAGE=VBSCRIPT %>
<% Option Explicit %>
```

pagina5

```
<HTML><HEAD><TITLE>Curso ASP</TITLE></HEAD>
<BODY>
<% Dim A, B, S, R(2)
A = Paulo Henrique <!--string-->
B = 200 <!--inteiro-->
S = #24/09/2001# <!--data-->
R(0) = Paulo Henrique <!--string dentro de array-->
R(1) = 200 <!--inteiro dentro de array-->
R(2) = #24/09/2001# <!--data dentro de array-->
%>
nome = <%= A%><BR>
número = <%= B%><BR>
data = <%= S%><BR>
elemento 1 do vetor = <%= R(0)%><BR>
elemento 2 do vetor = <%= R(1)%><BR>
elemento 3 do vetor = <%= R(2)%><BR>
</BODY></HTML>
```

CONSTANTES

Uma constante representa um valor fixo através de um identificador alfanumérico. Uma vez definido, o valor não pode ser modificado.

Exemplo 6: const.asp:

```
<% @ LANGUAGE=VBSCRIPT %>
<% Option Explicit %>
<HTML><HEAD><TITLE>Curso ASP</TITLE></HEAD>
<BODY>
<% Const A
A = 560 <!--não pode mais ser modificado-->
%>
</BODY></HTML>
```

OPERADORES

Símbolos alfanuméricos para efetuar operações:

operador função

= atribuição / igualdade

<> diferença

< menor que

<= menor ou igual que

> maior que

>= maior ou igual que

+ soma / concatenação de strings

- subtração / negativo numérico

* multiplicação

/ divisão

\ divisão com resultado de número inteiro

Mod retorna o resto de uma divisão entre inteiros

^ exponenciação

& concatenação de strings

Is igualdade entre dois objetos

Obs.: Os valores envolvidos numa mesma operação devem ser do mesmo subtipo ou de subtipos compatíveis.

pagina 6

Exemplo 7: oper.asp:

```
<% @ LANGUAGE=VBSCRIPT %>
<% Option Explicit %>
<HTML><HEAD><TITLE>Curso ASP</TITLE></HEAD>
<BODY>
<% Dim a, b, f, g, s
a = "1"
b = "2"
f = 1
g = 2
s = "letra"
%>
operações numéricas:<BR>
F + G = <% =f+g %><BR> <!--soma-->
G - F = <% =g-f %><BR><BR> <!--diferença-->
concatenação de strings:<BR>
A + B = <% =a+b %><BR><BR> <!--string+string-->
junção de número com string:<BR>
G + A = <% =g+a %><BR> <!--número + string numérica-->
G + S = <% =g+s %> <!--incompatibilidade = número + string alfanumérica -->
</BODY></HTML>
```

Para resolver esses problemas, algumas conversões de tipos podem ser utilizadas:

Função Conversão

CStr expressão => string

Cint expressão => integer

CLng expressão => long

Cbool expressão => booleano

Cbyte expressão => byte

Cdate expressão => date

CDbl expressão => double

CSng expressão => single

Exemplo 8: conver.asp:

```
<% @ LANGUAGE=VBSCRIPT %>
<% Option Explicit %>
<HTML><HEAD><TITLE>Curso ASP</TITLE></HEAD>
<BODY>
<% Dim a, b, aniver, hoje
a = "1" : b = "2"
aniver = #12/03/1992#
hoje = Date
%>
Operações:<BR>
A - B = <% =Cint(a)-Cint(b) %><BR>
A + B = <% =Cint(a)+Cint(b) %><BR><BR>
Idade:<BR>
Eu tenho <% =CLng(hoje-aniver) %> dias de vida<BR>
Farei 10000 dias de vida em <% =Cdate(aniver+10000) %>
</BODY></HTML>
```

SUBROTINAS

Pagina7

VBScript possui dois tipos de subrotinas: Sub e Function. Um Sub é um conjunto de comandos associados a um identificador alfanumérico. Uma Function possui a mesma definição só que além de executar os comandos a ela associados, pode gerar um valor como resultado. Essas subrotinas podem ainda receber algum(s) valor(s) como parâmetro. Suas respectivas sintaxes:

```
Sub nome_procedimento([parâmetro(s)])  
comando(s)  
End Sub  
Function nome_função([parâmetro(s)])  
comando(s)  
End Function
```

Exemplo 9: rotinas.asp:

```
<% @ LANGUAGE=VBSCRIPT %>  
<% Option Explicit %>  
<HTML><HEAD><TITLE>Curso ASP</TITLE></HEAD>  
<BODY>  
<% Function Soma(a,b)  
Dim Resultado  
Resultado = a + b  
Soma = Resultado  
end Function %>  
<% Dim e  
e = Soma(10,20) %>  
<% Dim A,B  
A = "Variável Global A"  
B = "Variável Global B"  
MudaB %>  
Resultados da Sub:<BR>  
Valor de A = <%=A%><BR>  
Valor de B = <%=B%><BR><BR>  
<% Sub MudaB()  
Dim A  
A = "Variável A no SUB"  
B = "Variável B no SUB"  
End Sub  
%>  
Resultados da Function:<BR>  
Soma(10,20)= <%= e%><BR>  
Soma(100,200)=<%=Soma(100,200) %>  
</BODY></HTML>
```

Observe que o código para o SUB (ou Function) não precisa ser escrito antes de sua chamada.

INCLUDE FILES

Essa é mais uma das formas que existe para poupar trabalho dos programadores economizando linhas de código. A idéia é criar um arquivo texto de qualquer extensão que contenha um conjunto de subrotinas. Essas, estarão disponíveis a qualquer página asp que faça referência a esse arquivo. Existem duas formas de referenciar tal arquivo numa página ASP:

```
<!-- #INCLUDE VIRTUAL="path_virtual/nome_arquivo" --> ou  
<!-- #INCLUDE FILE="path_fisico/nome_arquivo" -->
```


FUNÇÕES

Função Retorno

Abs (valor_numérico) módulo de um número

Fix (valor_numérico) parte inteira de um número

Int (valor_numérico)

parte inteira de um número

se valor < 0, imediatamente menor

Log (valor_numérico) logaritmo Neperiano de um número

Sqr (valor_numérico) raiz quadrada de um valor numérico

Date data atual

Time hora atual

Now data e hora atual

Day (valor_data) dia de uma determinada data

Month (valor_data) mês de uma determinada data

Year (valor_data) ano de uma determinada data

Weekday (valor_data)

dia da semana no formato numérico de

uma determinada data

Hour (tempo)

hora de uma determinada expressão

de tempo

Minute (tempo)

minutos de uma determinada

expressão de tempo

Second (tempo)

segundos de uma determinada

expressão de tempo

Asc (caractere) correspondente ASCII do caracter

Lcase (string) todos os caracteres minúsculos

Ucase (String) todos os caracteres maiúsculos

Len (String) número de caracteres de uma string

FormatCurrency (Valor Numérico) padrão moeda configurado no servidor