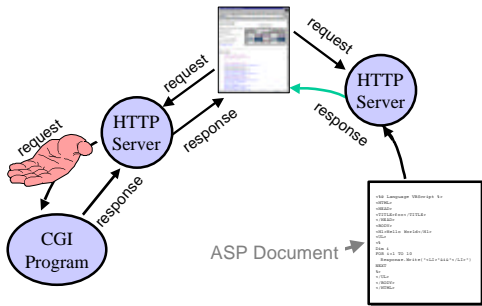


Active Server Pages

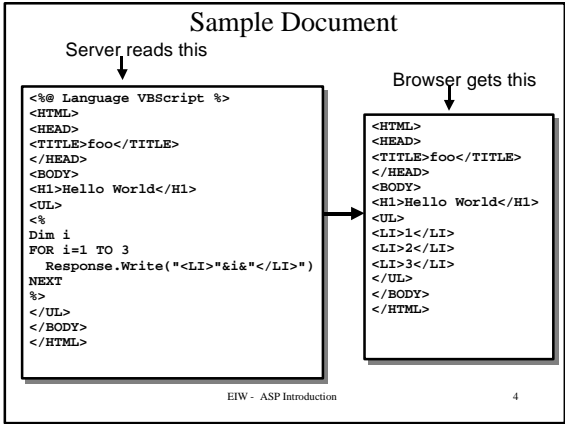
VBScript

CGI vs. "Smart Server"



Active Server Pages

- The document contains some commands that the server processes.
 - Microsoft ASP supports
 - VBScript (similar to Visual Basic)
 - JScript (Javascript)
- There are other languages supported this way (by some HTTP Servers).
 - PHP, Cold Fusion, JSP



VBScript

- Another language to learn!
- Microsoft Only.
- Very Different Syntax (compared to C++, JavaScript, Perl). No semicolons!
- Objects
- The server provides lots of Web programming support (many tasks are *easier*).

EIW - ASP Introduction 5

VBScript Language

- Variables
 - variant
- Operators
 - some are the same, others different...
- Control Structures
 - same stuff, different syntax
- Subroutines & Functions
 - collectively called *procedures*
- Objects
 - lots of functionality provided by objects

EIW - ASP Introduction 6

VBScript is not case sensitive

- **foo** is the same thing as **FOO**
- This applies to everything
 - variable names, constant names
 - keywords
 - subroutine/function names
 - object names

EIW - ASP Introduction

7

VBScript Variables

- Data type: *variant*
 - can be anything (much like JavaScript & Perl).
- Some people name variables according to how they will be used:
 - **intFoo** Foo is an int
 - **strFoo** Foo is a string
 - **objFoo** Foo is an object

EIW - ASP Introduction

8

Variable Declarations

- Variables don't have to be declared, but they can be using **Dim**
Dim foo
- If you add the tag `<% Option Explicit %>` to the top of your document, the server will insist that all variables must be declared.
 - this is generally a good idea! Saves you from wasting time tracking down typos.

EIW - ASP Introduction

9

What's up with **Dim**?

- A holdover from the Basic programming language.
- Dim was for "Dimension" (used when declaring arrays).
- Send complaints to bill@microsoft.com
 - Bill wrote the original Basic interpreter used on PCs

EIW - ASP Introduction

10

Variable Scope

- Variables declared outside of a subroutine or function (VBScript has both) are *global*.
- Variables declared in a subroutine or function are created and destroyed each time the procedure is called. (*local variables*)
- You can also create Session variables and Application variables! (more about this later).

EIW - ASP Introduction

11

Dim-witted Example

```
<%  
Dim wit ← declares a variables  
Dim sum ← named wit and sum  
sum=0  
FOR wit=1 TO 3  
  Response.Write("<LI>" & wit & "</LI>")  
  sUm = SuM + WIT  
NEXT  
%> ← case doesn't matter  
</UL>  
<P>Total is <% =sum %>
```

EIW - ASP Introduction

12

VBScript Arithmetic Operators

Symbol	Operation
<code>^</code>	Exponentiation
<code>-</code>	Subtraction (negation)
<code>*</code>	Multiplication
<code>/</code>	Division
<code>\</code>	Integer Division
<code>Mod</code>	Modulus
<code>+</code>	Addition
<code>&</code>	String Concatenation

EIW - ASP Introduction

13

VBScript Comparison Operators

Symbol	Operation
<code>=</code>	Equality
<code><></code>	Inequality
<code><</code>	Less than
<code>></code>	Greater than
<code><=</code>	Less than or equal to
<code>>=</code>	Greater than or equal to
<code>Is</code>	Object equivalence

EIW - ASP Introduction

14

Assignment vs. Equality

- The assignment operator is the same as the logical equals comparison operator.

`a=3` ← This is assignment

If `a=3` Then
 `b=7`
End If ← This is comparison

EIW - ASP Introduction

15

VBScript Logical Operators

Symbol	Operation
Not	Logical Negation
And	Logical Conjunction
Or	Logical Disjunction
Xor	Logical Exclusion
Eqv	Logical Equivalence
Imp	Logical Implication

EIW - ASP Introduction

16

Logical Example

```
If a=3 or s="Hello" Then
    Response.Write("<B>BLAH</B>")
End If
```

EIW - ASP Introduction

17

Control Structures

- In VBScript they are called "Statements"
- **If** and friends
- Loops: **For**, **For Each**, **While**, **Do**
- Exception Handling: **On Error**
- **Sub** and **Function**
- **Select**
- Lots more...

EIW - ASP Introduction

18

If Statements

- Inline (all on one line)

```
If i > 3 Then a = 20 Else b = 30
```

- Block

```
If i > 3 Then
```

```
  a = 20
```

```
Else
```

```
  b = 30
```

```
End If
```

keywords in blue

EIW - ASP Introduction

19

Fancy Inline Ifs

```
If A > 12 Then a=3 : b=4 : c=5
```



colon between each statement

This must be all on one line!

EIW - ASP Introduction

20

If Then ElseIf Else

```
If A > 3 And s = "Sam" Then
```

```
  Asp = "fun-fun-fun"
```

```
ElseIf A > 3 Then
```

```
  Asp = "fun-fun"
```

```
ElseIf A > 1 Then
```

```
  Asp = "fun"
```

```
Else
```

```
  Asp = "I want Perl back"
```

```
End If
```

EIW - ASP Introduction

21

For loops

```
For counter=1 To 100
  sum = sum + counter
Next

For counter=10 To 1 Step -1
  sum = sum + counter
Next
```

EIW - ASP Introduction

22

Quick Exit

```
For j=1 To 9
  If x[j]=-1 Then
    Exit For
  End If
  sum = sum + j
Next
```

EIW - ASP Introduction

23

While loop

```
i=0
While i<0
  sum = sum + i
  i = i + 1
Wend
```

EIW - ASP Introduction

24

Do loops

```
Do While condition      Do
  . . .                 . . .
Loop                    Loop While condition

Do Until condition     Do
  . . .                 . . .
Loop                    Loop Until condition
```

EIW - ASP Introduction

25

Do Examples

```
i=0
Do While i<10
  sum = sum + i
  i = i + 1
Loop

Do
  sum = sum - i
  i = i - 1
Loop Until i=0
```

EIW - ASP Introduction

26

Do quick exit

```
Do While i<10
  grade = grade + hw[i]
  If sum > 100
    Exit Do
  End If
Loop
Response.Write("HW Grade: " & grade)
```

EIW - ASP Introduction

27

Exceptions Handling (Errors)

- Run time errors will result in the server sending error information to the browser (as part of the content).
 - This is great for debugging!
 - This is bad for production systems.
 - generally want to handle errors ourselves instead of having the server send a cryptic message to the user.

EIW - ASP Introduction

28

VBScript On Error

- You can tell VBScript what to do when a runtime error happens.
 - Each procedure can handle errors itself, or allow calling procedure to handle things.
 - Sometimes you just want to go on as if nothing happened.
 - Sometimes you need to handle errors with special code.

EIW - ASP Introduction

29

Error Example

```
i=3
While i>-2
  j=10/i
  Response.Write("j is "&j"&"<BR>")
  i=i-1
Wend
```

```
j is 3.3333333333333333
j is 5
j is 10
Microsoft VBScript runtime error '800a000b'
Division by zero
/eiw/asp/err.asp, line 8
```

EIW - ASP Introduction

30

On Error Resume Next

- If you want to continue on as if nothing happened (usually not a good idea!).
- Execution continues at the statement after the one that is causing problems.
- Sets error handling until the end of a procedure (or another **On Error** statement).

EIW - ASP Introduction

31

On Error Resume Next Example

```
On Error Resume Next
i=3
While i>-2
  j=10/i
  Response.Write("j is "&j"<BR>")
  i=i-1
Wend
```

Old value of j! →

```
j is 3.3333333333
j is 5
j is 10
j is 10
j is -10
```

EIW - ASP Introduction

32

Better Example

```
On Error Resume Next
i=3
While i>-2
  j=10/i
  If Err.Number > 0 Then
    Response.Write("j is a problem<BR>")
    Err.Clear
  Else
    Response.Write("j is "&j"<BR>")
  End If
  i=i-1
Wend
```

```
j is 3.3333333333
j is 5
j is 10
j is a problem
j is -10
```

EIW - ASP Introduction

33

On Error GoTo 0

- You can turn off "On Error Resume Next" with the statement:
`On Error Goto 0`
- Stays off until end of procedure or another `On Error Resume Next` statement.

EIW - ASP Introduction

34

Response.Write()

- **Response** Object
 - corresponds to the HTTP response
 - Method `Write()` sends text to the browser

```
Response.Write("<H1>Welcome</H1>")
```

EIW - ASP Introduction

35

Table of Squares

```
<TABLE>
<TR><TH>Num</TH><TH>Num Squared</TH></TR>
<%
Dim i
For i=1 To 10
    Response.Write("<TR><TD>"&i"</TD>")
    Response.Write("<TD>"&i^2"</TD></TR>")
Next
%>
</TABLE>
```

EIW - ASP Introduction

36

Sending a single expression

- You can include the value of a single expression without `Response.Write()`

```
<%=i%>  
<%=i+j%>  
<%=Request("fname")%>
```

EIW - ASP Introduction

37

Another way to do the Table of Squares

```
<% Dim i %>  
<TABLE>  
<TR><TH>Num</TH><TH>Num Squared</TH></TR>  
<% For i=1 To 10 %>  
<TR><TD><%=i%></TD><TD><%=i^2%></TD></TR>  
<% Next %>  
</TABLE>
```

EIW - ASP Introduction

38
