

VBScript Variables

Variable is a named memory location used to hold a value that can be changed during the script execution. VBScript has only **ONE** fundamental data type, **Variant**.

Rules for Declaring Variables:

- Variable Name must begin with an alphabet.
- Variable names cannot exceed 255 characters.
- Variables Should NOT contain a period.
- Variable Names should be unique in the declared context.

Declaring Variables

Variables are declared using “dim” keyword. Since there is only ONE fundamental data type, all the declared variables are variant by default. Hence, a user **NEED NOT** mention the type of data during declaration.

Example 1: In this Example, IntValue can be used as a String, Integer or even arrays.

```
Dim Var
```

Example 2: Two or more declarations are separated by comma,

```
Dim Variable1,Variable2
```

Assigning Values to the Variables

Values are assigned similar to an algebraic expression. The variable name on the left hand side followed by an equal to = symbol and then its value on the right hand side.

Rules:

- The numeric values should be declared without double quotes.
- The String values should be enclosed within doublequotes "
- Date and Time variables should be enclosed within hash symbol #

Examples:

```
' Below Example, The value 25 is assigned to the variable.
Value1 = 25

' A String Value 'VBScript' is assigned to the variable StrValue.
StrValue = "VBScript"

' The date 01/01/2020 is assigned to the variable DToday.
Date1 = #01/01/2020#

' A Specific Time Stamp is assigned to a variable in the below example.
Time1 = #12:30:44 PM#
```

Scope of the Variables

Variables can be declared using the following statements that determines the scope of the

variable. The scope of the variable plays a crucial role when used within a procedure or classes.

- Dim
- Public
- Private

Dim

Variables declared using “Dim” keyword at a Procedure level are available only within the same procedure. Variables declared using “Dim” Keyword at script level are available to all the procedures within the same script.

Example : In the below example, the value of Var1 and Var2 are declared at script level while Var3 is declared at procedure level.

Note : The scope of this chapter is to understand Variables. Functions would be dealt in detail in the upcoming chapters.

```
<!DOCTYPE html>
<html>
<body>
<script language="vbscript" type="text/vbscript">

Dim Var1
Dim Var2

Call add()
Function add()
    Var1 = 10
    Var2 = 15
    Dim Var3
    Var3 = Var1+Var2
    MsgBox Var3 'Displays 25, the sum of two values.
End Function

Msgbox Var1    ' Displays 10 as Var1 is declared at Script level
Msgbox Var2    ' Displays 15 as Var2 is declared at Script level
Msgbox Var3    ' Var3 has No Scope outside the procedure. Prints Empty

</script>
</body>
</html>
```

Public

Variables declared using "Public" Keyword are available to all the procedures across all the associated scripts. When declaring a variable of type "public", Dim keyword is replaced by "Public".

Example : In the below example, Var1 and Var2 are available at script level while Var3 is available across the associated scripts and procedures as it is declared as Public.

```
<!DOCTYPE html>
<html>
<body>
<script language="vbscript" type="text/vbscript">
Dim Var1
Dim Var2
Public Var3

Call add()

Function add()
    Var1 = 10
    Var2 = 15
    Var3 = Var1+Var2
```

```

    MsgBox Var3 'Displays 25, the sum of two values.
End Function

Msgbox Var1    ' Displays 10 as Var1 is declared at Script level
Msgbox Var2    ' Displays 15 as Var2 is declared at Script level
Msgbox Var3    ' Displays 25 as Var3 is declared as Public

</script>
</body>
</html>

```

Private

Variables that are declared as "Private" have scope only within that script in which they are declared. When declaring a variable of type "Private", Dim keyword is replaced by "Private".

Example : In the below example, Var1 and Var2 are available at Script Level. Var3 is declared as Private and it is available only for this particular script. Use of "Private" Variables is more pronounced within the Class.

```

<!DOCTYPE html>
<html>
<body>
<script language="vbscript" type="text/vbscript">
Dim Var1
Dim Var2
Private Var3

Call add()
Function add()
    Var1 = 10
    Var2 = 15
    Var3 = Var1+Var2
    MsgBox Var3 'Displays the sum of two values.
End Function

Msgbox Var1    ' Displays 10 as Var1 is declared at Script level
Msgbox Var2    ' Displays 15 as Var2 is declared at Script level
Msgbox Var3    ' Displays 25 but Var3 is available only for this script.

</script>
</body>
</html>

```

Processing math: 100%