



CX's VBScript Diary

[<Visual Studio は VBScript 用の最強エデ...](#)

2010-01-31

VBScript 基礎文法最速マスター ★★

11:50 |  43 users

VBScript の文法一覧です。他の言語をある程度知っている人はこれを読めば VBScript の基礎をマスターして VBScript を書くことができるようになっていきます。簡易リファレンスとしても利用できると思いますので、これは足りないと思うものがあれば教えてください。

1. 基礎

変数宣言の強制

ソースコードの先頭に「Option Explicit」を入れるようにしましょう。変数の宣言が強制されコードの品質も上がります。

Option Explicit

メッセージの表示

```
WScript.Echo "Hello, VBScript World." ' コンソールにメッセージを出力します
MsgBox "Hello, VBScript World." ' メッセージボックスにメッセージを表示します
```

コメント

コメントは「'」を用います。「Rem」も使用可能です。

```
' コメントその1
```

```
Rem コメントその2
```

変数の宣言

変数の宣言は「Dim」を用います。固定サイズ配列の場合は「(10)」のようにサイズを指定します。サイズを指定しない場合は動的配列となります。

VBScript の変数自体はバリエーション型 (Variant) で代入する値に応じて内部形式が変化します。

```
Dim num ' 変数
Dim students(10) ' 配列変数 (配列は0~10となり要素数は11になります)
Dim students() ' 動的配列
```

スクリプトの実行

スクリプトを実行するにはコマンドラインで次のようにします。

```
CScript script.vbs
```

出力結果をファイルに書き出すにはリダイレクトを使います。

```
CScript script.vbs > file.txt
```

デバッガの起動

デバッガを起動するにはコマンドラインで次のようにします。

ただし、事前に Visual Studio 等のデバッガがインストールされている必要があります。

```
CScript script.vbs //X
```

2. 数値

数値の表現

変数には整数でも小数でも代入できます。代入する値に応じて、変数の内部形式が変化します。

```
num = 1           ' 整数型 (Integer)
num = 1.234       ' 倍精度浮動小数点型 (Double)
num = 1000000000 ' 長整数型 (Long)
```

四則演算

四則演算です。

```
num = 1 + 1      ' 2
num = 1 - 1      ' 0
num = 1 * 2      ' 2
num = 1 / 2      ' 0.5
```

商と余りの求め方です。

```
num = 3 \ 2      ' 1 (商)
num = 3 Mod 2    ' 1 (余り)
```

インクリメントとデクリメント

インクリメントとデクリメントです。

```
' インクリメント
i = i + 1

' デクリメント
i = i - 1
```

3. 文字列

文字列の表現

文字列は「"」ダブルクォーテーションで囲みます。変数にタブや改行コードをセットするには「vbTab」や「vbCrLf」を用います。

```
str1 = "abc"
```

```
str2 = "a" & vbTab & "bc" & vbCrLf
```

文字列操作

各種文字列操作です。

' 結合

```
str1 = "aaa" & "bbb"
```

' 文字列の連結

```
str2 = Join( Array("aaa", "bbb", "ccc"), ",")
```

' 区切り文字に「,」（カンマ）を指定する場合

' 分割

```
record = Split("aaa,bbb,ccc", ",")
```

' 長さ

```
length = Len("abcdef")
```

' 6

```
length = Len("あいうえお")
```

' 5 (文字数を数えるには Len 関数を使用します)

```
length = LenB("あいうえお")
```

' 10 (文字のバイト数を数えるには LenB 関数を使用します)

' 切り出し

```
str = Mid("abcd", 1, 2)
```

' ab (1桁目から2文字)

' 検索

```
result = InStr("abcd", "cd")
```

' 見つかった場合はその位置、見つからなかった場合は 0 が返

```
result = InStr("あいうえお", "うえ")
```

' 3 (文字数で扱う場合は InStr 関数を使用します)

```
result = InStrB("あいうえお", "うえ")
```

' 5 (バイト数で扱う場合は InStrB 関数を使用します)

4. 配列

配列変数の宣言と代入

固定サイズ配列として宣言する場合

```
Dim ary(2)
```

```
ary(0) = 100
```

```
ary(1) = 200
```

```
ary(2) = 300
```

動的配列として宣言する場合

```
Dim ary()
```

```
ReDim ary(2)
```

```
ary(0) = 100
```

```
ary(1) = 200
```

```
ary(2) = 300
```

変数に Array 関数を使用して配列をセットする場合

```
Dim ary
```

```
ary = Array( 100, 200, 300 )
```

配列の要素の参照と代入

```
a = ary(0) ' 100
b = ary(1) ' 200
```

```
ary(0) = 1
ary(1) = 2
```

要素の個数

```
n = UBound(ary) - LBound(ary) + 1 ' 配列の上限 - 下限
```

配列の操作

```
Dim ary
ary = Array( 1, 2, 3 )

' 先頭を取り出す
a = ary(0) ' a は 1
' 末尾を取り出す
b = ary(UBound(ary)) ' b は 3
' 末尾に追加
ReDim Preserve ary(UBound(ary) + 1) ' 固定サイズ配列の場合は追加できません
ary(UBound(ary)) = 9 ' ary は [1, 2, 3, 9] に
```

5. Dictionary オブジェクト

VBScript にはハッシュ変数はありませんが、Scripting.Dictionary オブジェクトを用いることで代替が可能です。

Dictionary オブジェクトの宣言と代入

```
Dim hash
Set hash = CreateObject("Scripting.Dictionary")
hash.Add "a", 1
hash.Add "b", 2
```

Dictionary の要素の参照と代入

```
' 要素の参照
WScript.Echo hash("a") ' 1
WScript.Echo hash("b") ' 2

' 要素の代入
hash("a") = 5
hash("b") = 7
```

Dictionary のプロパティとメソッド

```
' キーの取得
keys = hash.Keys
```

```
' 値の取得
```

```
values = hash.Items
```

```
' キーの存在確認
```

```
hash.Exists("a")
```

```
' キーの削除
```

```
hash.Remove "a"
```

6. 制御文

If文

If文です。1行に書く形式とブロック形式の構文が利用できます。

```
If 条件 Then 式 [Else 式]
```

ブロック形式の If 文には End If が必要です。

```
If 条件 Then
```

```
  式
```

```
End If
```

If ~ Else 文

If ~ Else文です。

```
If 条件 Then
```

```
  式
```

```
Else
```

```
  式
```

```
End If
```

If ~ ElseIf 文

If ~ ElseIf文です。「Else If」ではなく「ElseIf」である(Else と If の間に空白は入らない)ことに注意しましょう。

```
If 条件 Then
```

```
  式
```

```
ElseIf 条件 Then
```

```
  式
```

```
End If
```

Do ~ Loop 文

Do ~ Loop 文です。Do While ~ (真の間ループ) や Do Until ~ (真になるまでループ) が利用できます。

```
i = 0
```

```
Do While i < 5
```

```
  ' 処理
```

```
i = i + 1  
Loop
```

For 文

For文です。

```
For i = 0 To 4
```

```
Next
```

For Each 文

For Each 文です。配列やコレクションオブジェクトの各要素を参照するときに便利です。

```
For Each field In fields
```

```
Next
```

7. サブルーチン

Sub プロシージャ

戻り値を返さない処理は Sub プロシージャで定義します。

```
Sub show_sum( num1, num2 )  
    Dim total  
    total = num1 + num2  
    WScript.Echo total  
End Sub
```

Function プロシージャ

戻り値を返す処理は Function プロシージャで定義します。

```
Function sum( num1, num2 )  
    Dim total  
    total = num1 + num2  
    sum = total ' 戻り値を指定  
End Function
```

8. ファイル入出力

VBScript でファイルの入出力を行うには Scripting.FileSystemObject オブジェクトを使用します。

```
Const ForReading = 1  
  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set file = fso.OpenTextFile("C:¥temp¥hoge.txt", ForReading)  
Do Until file.AtEndOfStream  
    strLine = file.ReadLine ' ファイルの 1 行分を読み込みます  
Loop
```

```
file.Close
```

9. 知っておいたほうがよい文法

VBScript でよく出てくる知っておいたほうがよい文法の一覧です。

オブジェクトの代入と破棄

オブジェクト変数に代入する場合は Set 文を使用します。

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

オブジェクト変数を破棄する場合は Set 文で Nothing をセットします。

```
Set fso = Nothing
```

Null と Empty と Nothing の違い

Null … 変数に有効な値が変数に格納されていないことを示します。DB の Null 値に相当します。

Empty … 変数が初期化されていない状態を表す値です。変数を宣言した直後はこの値です。

Nothing … オブジェクト変数がオブジェクトを参照していない状態を表す値です。

VBScript の真偽値

VBScript で偽と判断される値は、「False」「0」「Null」「Empty」「False”」「"0"」です。これ以外は真になります。

ByVal と ByRef

VBScript の引数には、値渡し(ByVal)と参照渡し(ByRef)があります。省略した場合は参照渡しになります。

```
Dim num1, num2, num3
num1 = 0
num2 = 0
num3 = 0

test num1, num2, num3

WScript.Echo num1 ' 1
WScript.Echo num2 ' 0
WScript.Echo num3 ' 3

Sub test( num1, ByVal num2, ByRef num3 )
    num1 = 1 ' 参照渡しの為、値は更新されます
    num2 = 2 ' 値渡しの為、値は更新されません
    num3 = 3 ' 参照渡しの為、値は更新されます
End Sub
```

While ~ Wend 文

While ~ Wend 文です。Do ~ Loop 文の方が柔軟性がありますが、覚えておいて損はありません。

```
i = 0
While i < 5
```

```
' 処理
```

```
    i = i + 1  
Wend
```

コマンドライン引数

WScript.Arguments でコマンドラインの情報を取得できます。

また、Arguments オブジェクトの Named プロパティを使用することで名前付き引数が利用可能です。

```
For i = 0 To WScript.Arguments.Count - 1  
    WScript.Echo WScript.Arguments(i)  
Next
```

エラー処理

VBScript では、実行時にエラーがあると処理が停止します。

処理を停止させず続行させる場合には On Error Resume Next を用います。エラー内容は Err オブジェクトで判断します。

```
On Error Resume Next  
num = 1 / 0 ' 0 除算  
WScript.Echo "Err.Source      = [" & Err.Source & "]"      ' Microsoft VBScript 実行時エラー  
WScript.Echo "Err.Description = [" & Err.Description & "]"  ' 0 で除算しました。  
WScript.Echo "Err.Number     = [" & Err.Number & "]"      ' 11
```

インタラクティブVBScript

WSH は残念ながら対話型シェルの機能はありませんが、以下の [10 行ほどのプログラム](#) を用いることで VBScript を対話的に実行することができるようになります。

```
File : ivb.vbs  
Usage : CScript ivb.vbs  
Do While True  
    WScript.StdOut.Write(">>> ")  
    In = Wscript.StdIn.ReadLine  
    If LCase(Trim(In)) = "exit" Then Exit Do  
    On Error Resume Next  
    Err.Clear  
    Execute In  
    If Err.Number <> 0 Then WScript.Echo(Err.Description)  
    On Error Goto 0  
Loop
```

コマンドラインより「CScript ivb.vbs」と入力することで実行できます。終了は「exit」です。

```
C:\Users\cx20\edu\VBScript\ivb>CScript ivb.vbs  
Microsoft (R) Windows Script Host Version 5.7  
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```



```
>>> WScript.Echo "Hello"  
Hello  
>>> WScript.Echo 1+1  
2  
>>> exit
```

その他の基礎文法マスター

この記事は他の基本文法マスターに便乗して書いた物です。誤り等ございましたらコメント等して頂けると助かります。

- [Perl基礎文法最速マスター - Perl入門～サンプルコードによるPerl入門～](#)
- [Route 477 - Ruby基礎文法最速マスター](#)
- [PHP基礎文法最速マスター | Shin x blog](#)
- [Python基礎文法最速マスター - D++のはまり日誌](#)
- [Java基礎文法最速マスター - 何かしらの言語による記述を解析する日記](#)
- [VBA基礎文法最速マスター - 何かしらの言語による記述を解析する日記](#)
- [Brainf*ck基礎文法最速マスター - 医者を目指す妻を応援する夫の日記](#)
- [Haskell基礎文法最速マスター - think and error](#)
- [Bash基礎文法最速マスター - 何かしらの言語による記述を解析する日記](#)
- [VBScript 基礎文法最速マスター - CX's VBScript Diary - VBScript グループ](#)
- [JavaScript基礎文法最速マスター - なんとなく日記](#)

[コメントを書く](#)

[<Visual Studio は VBScript 用の最強エデ...](#)