

# DOS/VBScript

「△」は、半角スペースです。

## ■DOS

### 標準出力と標準エラー出力

コマンドを実行した時にエラーが発生した場合には標準出力ではなく標準エラー出力に対して出力が行われます。

#### 1、ファイルの拡張子(.bat or .cmd)

2、各コマンドのオプションは、[オプション名]△/?または[オプション]△-helpで確認できます。

#### 3、一部だけ紹介

##### (1) コメント

```
rem△[コメント]
```

(例)

```
@echo△off
```

```
rem△このプログラムは○○の処理を行います。
```

##### (2) echo

```
echo△[メッセージ]・・・メッセージを表示
```

##### (3) dir

ディレクトリ一覧

##### (4) pause

バッチファイルの処理を一時停止させる

##### (5) cd

```
cd△[ディレクトリ名]・・・カレントディレクトリを移動
```

##### (6) mkdir

```
mkdir△[ディレクトリ名]・・・ディレクトリ新規作成
```

##### (7) rmdir

```
rmdir△[ディレクトリ名]・・・ディレクトリの削除
```

## (8) del

del△[ファイル名]・・・ファイルの削除

## (9) copy・・・ファイルのコピー

## (10) xcopy・・・copy コマンドではできない、サブディレクトリを含めたディレクトリ構造とファイルの一括コピー

## (11) rename

rename△[変更前]△[変更後]・・・ファイル/ディレクトリの名前変更

## (12) move

move△[移動前]△[移動後]・・・ファイル/ディレクトリの移動

## (13) shutdown

コンピュータの停止や再起動。リモートのコンピュータに対して行うことができる。

## (14) net start

net△start△[サービス名]・・・サービスの起動

## (15) net stop

net△stop△[サービス名]・・・サービスの停止

## (16) 引数

1つの目の引数が「%1」。2つ目の引数が「%2」。

## (17) if

(例)

```
if△exist△C:¥a.txt△(type△C:¥a.txt
  echo△type△OK
)△else△(
  echo△type△NG
)
```

## 5、VBScript ファイルの呼び出し方法

```
cscript△[VBScript 名].vbs△C:¥temp¥test.log△2>&1
```

「2>&1」と書いた場合、標準出力と標準エラー出力を出力します。

## 6、何か実行の戻り値確認

%ERRORLEVEL%の値を見る。

(例)

```
if△%ERRORLEVEL%==0△echo△“問題ない”
```

## 7、環境変数に値を設定

(例)

```
SET△FILE_NAME=temp.log
```

## 8、ファイル出力(上書/追記)

```
dir△>△[上書きするファイル]
```

```
dir△>>△[追記するファイル]
```

### <おまけ：DOS コマンド(全部かどうかは不明)>

- ASSOC . . . ファイル拡張子の関連付けを表示または変更します。
- AT . . . コマンドやプログラムを指定した日時に実行します。
- ATTRIB . . . ファイルの属性を表示または変更します。
- BREAK . . . 拡張 CTRL+C チェックを設定または解除します。
- CACLS . . . ファイルのアクセス制御リスト (ACL) を表示または変更します。
- CALL . . . バッチ ファイル中から、別のバッチ ファイルを呼び出します。
- CD . . . 現在のディレクトリを表示または変更します。
- CHCP . . . 有効なコード ページ番号を表示または設定します。
- CHDIR . . . 現在のディレクトリを表示または変更します。
- CHKDSK . . . ディスクをチェックし、状態を表示します。
- CHKNTFS . . . 起動時のディスクのチェックを表示または変更します。
- CLS . . . 画面を消去します。
- CMD . . . Windows コマンド インタープリタを新しく起動します。
- COLOR . . . コンソールの文字と背景の既定の色を設定します。
- COMP . . . 2 個のファイルまたはファイルの集合の内容を比較します。
- COMPACT . . . NTFS パーティション上のファイルの圧縮状態を表示または変更します。
- CONVERT . . . FAT ボリュームを NTFS に変換します。現在のドライブは変換できません。

COPY . . . 1 個以上のファイルを別の場所にコピーします。

DATE . . . 日付を表示または変更します。

DEL . . . 1 個以上のファイルを削除します。

DIR . . . ディレクトリ中のファイルやサブディレクトリの一覧を表示します。

DISKCOMP . . . 2 つのフロッピー ディスクの内容を比較します。

DISKCOPY . . . フロッピー ディスクの内容を別のフロッピー ディスクにコピーします。

DOSKEY . . . コマンド ラインの編集、Windows コマンドの再呼び出し、マクロの作成。

ECHO . . . メッセージの表示、コマンド エコーのオン、オフの指定をします。

ENDLOCAL . . . バッチ ファイルで、環境変数のローカライズを終了します。

ERASE . . . 1 個以上のファイルを削除します。

EXIT CMD.EXE . . . プログラム (コマンド インタープリタ) を終了します。

FC . . . 2 個のファイルまたはファイルの集合の内容を比較して、それらの違いを表示。

FIND . . . ファイルの中からテキスト文字列を検索します。

FINDSTR . . . ファイルの中から文字列を検索します。

FOR . . . 指定されたコマンドを、ファイルの集合の各ファイルに対して実行します。

FORMAT . . . Windows で使用するためのディスクをフォーマットします。

FTYPE . . . ファイル拡張子の関連付けで使われるファイル タイプを表示または変更します。

GOTO . . . バッチ ファイル中で、ラベルで定義されている行へ 実行を移します。

GRAFTABL . . . Windows がグラフィック モードで拡張文字セットを表示できるようにします。

HELP . . . Windows コマンドのヘルプ情報を表示します。

IF . . . バッチ ファイル中で、条件処理を実行します。

LABEL . . . ディスクのボリューム ラベルを作成、変更、または削除します。

MD . . . ディレクトリを作成します。

MKDIR . . . ディレクトリを作成します。

MODE . . . システム デバイスを設定します。

MORE . . . 出力を一度に 1 画面ずつ表示します。

MOVE . . . 1 個以上のファイルをディレクトリから別のディレクトリに移動します。

PATH . . . 実行可能ファイルの検索パスを表示または設定します。

PAUSE . . . バッチ ファイルの処理を一時停止し、メッセージを表示します。

POPD . . . 現在のディレクトリを PUSHHD で保存したディレクトリに戻します。

PRINT . . . テキスト ファイルを印刷します。

PROMPT . . . Windows コマンド プロンプトを変更します。

PUSHHD . . . 現在のディレクトリを保存して、変更します。

RD . . . ディレクトリを削除します。

RECOVER . . . 不良または欠陥ディスクから読み出し可能な情報を復元します。

REM . . . バッチ ファイルや CONFIG.SYS の中で、コメント (注釈) を記録します。

REN . . . ファイルの名前を変更します。

RENAME . . . ファイルの名前を変更します。

REPLACE . . . ファイルを置き換えます。

RMDIR . . . ディレクトリを削除します。

SET . . . Windows 環境変数を表示、設定、または削除します。

SETLOCAL . . . バッチ ファイルで、環境変数のローカライズを開始します。

SHIFT . . . バッチ ファイルで、置き換え可能パラメータの位置をシフトします。

SORT . . . 入力を並べ替えます。

START . . . 別のウィンドウを起動して、指定したプログラムまたはコマンドを実行します。

SUBST . . . パスをドライブ名で置き換えます。

TIME . . . システム時刻を表示または変更します。

TITLE . . . コマンド プロンプト ウィンドウのタイトルを設定します。

TREE . . . ドライブまたはパスのディレクトリ構造を図式表示します。

TYPE . . . テキスト ファイルの内容を表示します。

VER . . . Windows のバージョンを表示します。

VERIFY . . . ファイルがディスクへ正しく書き込まれたかを照合するかどうか Windows へ指定します。

VOL . . . ディスクのボリューム ラベルとシリアル番号を表示します。

XCOPY . . . ファイルやディレクトリ構造をコピーします。

## ■VBScript

### 1、ファイルの拡張子(.vbs)

### 2、コメント

' コメント・・・シングルコーテーションの後ろがコメントになる

### 3、変数

```
Dim△[変数名]
```

### 4、定数

```
Const△[定数名]△=△[値]
```

### 5、引数の数を確認

```
WScript.Arguments.Count
```

引数の値を取得する場合は「WScript.Arguments(x)」を使用する。

「x」は「0」から始まる。

### 6、戻り値設定

```
WScript.Quit(0)・・・戻り値に「0」を指定
```

### 7、If

```
Dim x, y
```

```
x = 5
```

```
y = 3
```

```
If x > y Then
```

```
    MsgBox x & "は" & y & "より大きい。"
```

```
Else
```

```
    MsgBox x & "は" & y & "以下だ。"
```

```
End If
```

### 8、For

```
Dim j, total
```

```
For j = 2 To 10 Step 2
```

```
    total = total + j
```

```
Next
```

## 9、Function

Function△[関数名]([ByVal / ByRef] 引数名)

Exit△Function . . . . 関数から抜ける

End△Function

## 10、サブルーチン

Sub△[サブルーチン名]([ByVal / ByRef] 引数名)

Exit△Sub . . . . サブルーチンから抜ける

End△Sub

## 11、変数宣言しないとエラーになる宣言

Option Explicit

## <サンプル>

### ・ zip ファイル解凍

Dim ZipFile, ExtractTo, fso, objShell, FilesInZip, objFolder

```
Const FOF_SILENT          = &H04      '進捗ダイアログを表示しない。
Const FOF_RENAMEONCOLLISION = &H08    'ファイルやフォルダ名が重複するときは「コピー～」のようなファイル名にリネームする。
Const FOF_NOCONFIRMATION  = &H10     '上書き確認ダイアログを表示しない（[すべて上書き]と同じ）。
Const FOF_ALLOWUNDO      = &H40     '操作の取り消し（[編集]-[元に戻す]や{ctrl}+{z}）を有効にする。
Const FOF_FILESONLY      = &H80     'ワイルドカードが指定された場合のみ実行する。
Const FOF_SIMPLEPROGRESS = &H100    '進捗ダイアログは表示するがファイル名は表示しない。
Const FOF_NOCONFIRMMKDIR = &H200    'フォルダ作成確認ダイアログを表示しない（自動で作成）。
Const FOF_NOERRORUI      = &H400    'コピーや移動ができなかった場合の実行時エラーを発生させない。
                                'ただし、対象のファイルを飛ばして処理を続けるわけではないことに注意。
Const FOF_NORECURSION    = &H1000   'サブフォルダ内のファイルはコピーしない（ただし、フォルダは作成される）。
```

'ZIP ファイル

ZipFile = "C:¥Users¥test¥Desktop¥test.zip"

'解凍先

ExtractTo = "C:¥Users¥test¥Desktop¥"

'解凍先のフォルダが無い場合は作成する

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

If Not fso.FolderExists(ExtractTo) Then

    fso.CreateFolder(ExtractTo)

End If

'ZIP ファイルを解凍して、全てのファイルを解凍先フォルダにコピーする。

Set objShell = WScript.CreateObject("Shell.Application")

Set FilesInZip = objShell.NameSpace(ZipFile).items

Set objFolder = objShell.NameSpace(ExtractTo)

If (Not objFolder Is Nothing) Then

    objFolder.CopyHere FilesInZip, FOF\_NOCONFIRMATION + FOF\_SILENT

End If

'オブジェクトの破棄

Set fso = Nothing

Set objFolder = Nothing

Set FilesInZip = Nothing

Set objShell = Nothing



<課題> . . . 答えは次のページ

・ **bat ファイル側の処理**

- ・ bat ファイルの先頭と後ろで、処理開始と処理終了後メッセージをログと画面に出力  
(日付と時間も)
- ・ ログは「C:¥temp¥kadai.log」に出力。最初は上書き。以降は追記で出力する。
- ・ bat ファイルから VBScript ファイルを呼び出す。  
VBScript の標準出力と標準エラー出力をログファイルに追記(呼び出し時に指定)。
- ・ VBScript ファイル呼出し後に %ERRORLEVEL% を参照し、0 以外の場合は、  
処理が異常終了したことをログに出力。0 の場合は正常終了したことをログに出力。

・ **VBScript の処理**

- ・ 引数は「解凍する zip ファイル(パス名+ファイル名)」
- ・ 引数が 1 つではない場合は、処理終了。戻り値「-1」。WScript.Echo でエラーメッセージを  
標準出力に出力する。
- ・ 引数が空文字(Len と Trim を使用)の場合は、処理終了。戻り値「-1」。WScript.Echo で  
エラーメッセージを標準出力に出力する。
- ・ 引数のファイルが存在しない場合は、処理終了。戻り値「-1」。WScript.Echo でエラーメッ  
セージを標準出力に出力する。
- ・ フォルダ(C:¥Temp¥sample)が無い場合は作成する。
- ・ フォルダ(C:¥Temp¥sample)がある場合は、フォルダ内のファイル/フォルダを削除する。
- ・ 引数の zip ファイルを C:¥Temp¥sample に解凍する。
- ・ 戻り値「0」で処理を終了。

## ■ datetime.bat

```
@echo off
set DATE_TMP=%DATE:/=%
set TIME_TMP=%TIME: =0%
set YYYY=%DATE_TMP:~0,4%
set YY=%DATE_TMP:~2,2%
set MM=%DATE_TMP:~4,2%
set DD=%DATE_TMP:~6,2%
set HH=%TIME_TMP:~0,2%
set MI=%TIME_TMP:~3,2%
set SS=%TIME_TMP:~6,2%
set SSS=%TIME_TMP:~9,2%
set DATETIME=%YYYY%%MM%%DD%%HH%%MI%%SS%%SSS%
set TIME_TMP=
set DATE_TMP=
```

## ■ kadai.bat

```
@echo off
rem 初期設定
set LOG_OUTPUT=C:¥temp¥kadai.log
set EXEC_VBS_FILE=C:¥temp¥kadai.vbs
set EXTRACT_FILE=C:¥temp¥kadai.zip
set DATETIME_BAT=C:¥temp¥datetime.bat
set START_MSG=***** ファイル解凍処理：開始 *****
set NORMAL_END_MSG=***** ファイル解凍処理：正常終了 *****
set ABNORMAL_END_MSG=***** ファイル解凍処理：異常終了 *****
set NORMAL_RETURNCODE=0
set ABNORMAL_RETURNCODE=-1
rem 処理開始メッセージ
call %DATETIME_BAT%
echo %START_MSG% %DATE% %TIME%
echo %START_MSG% %DATE% %TIME% > %LOG_OUTPUT%
rem 標準出力と標準エラー出力を両方ともログファイルに出力
rem タイトルロゴを出さない(/nologo)
cscript %EXEC_VBS_FILE% %EXTRACT_FILE% //nologo >> %LOG_OUTPUT% 2>&1
rem 戻り値のチェック
set RC=%ERRORLEVEL%
if %ERRORLEVEL%==%ABNORMAL_RETURNCODE% goto ABNORMAL_END
rem 正常終了処理
call %DATETIME_BAT%
echo %NORMAL_END_MSG% %DATE% %TIME%
echo %NORMAL_END_MSG% %DATE% %TIME% >> %LOG_OUTPUT%
pause
exit /b NORMAL_RETURNCODE
rem 異常終了処理
:ABNORMAL_END
call %DATETIME_BAT%
echo %ABNORMAL_END_MSG%(%RC%) %DATE% %TIME%
echo %ABNORMAL_END_MSG%(%RC%) %DATE% %TIME% >> %LOG_OUTPUT%
pause
exit /b %ERRORLEVEL%
```

## ■ kadai.vbs

Option Explicit

' ■ 定数

```
Const START_MSG = "==== ファイル解凍処理(VBScript) : 開始 ====="  
Const END_MSG = "==== ファイル解凍処理(VBScript) : 終了 ====="  
Const NORMAL_END = 0  
Const ABNORMAL_END = -1  
Const ERROR_MSG_001 = "引数の数が正しくありません。"  
Const ERROR_MSG_002 = "引数が指定されていません。"  
Const ERROR_MSG_003 = "引数で指定されたファイルが存在しません。"  
Const OUTPUT_PATH = "C:¥temp¥sample¥"
```

' ■ 変数

```
Dim strZipFile  
Dim objFso  
Dim objParam  
Set objParam = WScript.Arguments
```

' ■ 開始メッセージ出力

```
WScript.Echo START_MSG
```

' ■ 引数チェック

'引数が1つではない場合、戻り値「-1」でスクリプト終了。

'WScript.Echo でエラーメッセージを出力

```
If objParam.Count <> 1 Then  
    WScript.Echo ERROR_MSG_001  
    WScript.Echo END_MSG  
    WScript.Quit ABNORMAL_END  
End If
```

'引数が空文字の場合、WScript.Echo でエラーメッセージを出力

'戻り値「-1」でスクリプト終了。

```
strZipFile = objParam(0)  
If Len(Trim(strZipFile)) = 0 Then  
    WScript.Echo ERROR_MSG_002  
    WScript.Echo END_MSG  
    WScript.Quit ABNORMAL_END  
End If
```

'引数のファイルが存在しない場合は、WScript.Echo でエラーメッセージを出力

'戻り値「-1」でスクリプト終了。

```
Set objFso = WScript.CreateObject("Scripting.FileSystemObject")  
If objFso.FileExists(strZipFile) = False Then  
    Set objFso = Nothing  
    WScript.Echo ERROR_MSG_003
```

```

WScript.Echo END_MSG
WScript.Quit ABNORMAL_END
End If
'オブジェクトの破棄
Set objFso = Nothing

'-----
'■C:%temp%sample フォルダは、無ければ作成
' 引数の zip ファイルを C:%temp%sample に解凍
'-----

Call Extract(strZipFile, OUTPUT_PATH)

'-----
'■終了メッセージ出力
' スクリプト終了。終了コード：0
'-----

WScript.Echo END_MSG
WScript.Quit NORMAL_END

'-----
'関数名：Extract・・・Zip ファイル解凍処理
'引数：ZipFile・・・Zip ファイル名(フルパス)
'      : ExtractTo・・・Zip ファイル解凍先フォルダ名(末尾に「¥」)
'-----

Sub Extract(ZipFile, ExtractTo)
  Const FOF_SILENT          = &H04          '進捗ダイアログを表示しない。
  Const FOF_RENAMEONCOLLISION = &H08        'ファイルやフォルダ名が重複するときは「コピー～」のようなファイル名にリネームする。
  Const FOF_NOCONFIRMATION  = &H10        '上書き確認ダイアログを表示しない（[すべて上書き]と同じ）。
  Const FOF_ALLOWUNDO      = &H40        '操作の取り消し（[編集]-[元に戻す]や{ctrl}+{z}）を有効にする。
  Const FOF_FILESONLY      = &H80        'ワイルドカードが指定された場合のみ実行する。
  Const FOF_SIMPLEPROGRESS = &H100       '進捗ダイアログは表示するがファイル名は表示しない。
  Const FOF_NOCONFIRMMKDIR = &H200       'フォルダ作成確認ダイアログを表示しない（自動で作成）。
  Const FOF_NOERRORUI      = &H400       'コピーや移動ができなかった場合の実行時エラーを発生させない。
                                          'ただし、対象のファイルを飛ばして処理を続けるわけではないことに注意。
  Const FOF_NORECURSION    = &H1000      'サブフォルダ内のファイルはコピーしない（ただし、フォルダは作成される）。

  Dim objFso
  Dim objShell
  Dim FilesInZip
  Dim objFolder

  '解凍先フォルダが無い場合は作成
  '解凍先フォルダがある場合は、フォルダ内のファイル/フォルダを削除
  Set objFso = WScript.CreateObject("Scripting.FileSystemObject")
  If objFso.FolderExists(ExtractTo) = True Then
    Call DeleteFolder(ExtractTo)
  Else
    objFso.CreateFolder ExtractTo
  End If

```

```
'Zip ファイルを解凍して、全てのファイルを解凍先フォルダにコピー
Set objShell = WScript.CreateObject("Shell.Application")
Set FilesInZip = objShell.NameSpace(ZipFile).items
Set objFolder = objShell.NameSpace(ExtractTo)
If (Not objFolder Is Nothing) Then
    objFolder.CopyHere FilesInZip, FOF_NOCONFIRMATION + FOF_SILENT
End If
```

```
'オブジェクトの破棄
```

```
Set objFso = Nothing
Set objFolder = Nothing
Set FilesInZip = Nothing
Set objShell = Nothing
```

```
End Sub
```

---

```
'関数名 : DeleteFolder . . . 指定フォルダ内のファイル/フォルダ削除処理
```

```
'引数 : strFolder . . . フォルダ名(フルパス)
```

---

```
Sub DeleteFolder(strFolder)
```

```
Dim objFso
Dim objFolder
Dim file
Dim subfolder
```

```
Set objFso = WScript.CreateObject("Scripting.FileSystemObject")
Set objFolder = objFso.GetFolder(strFolder)
```

```
'ファイルを列挙
```

```
For Each file In objFolder.Files
    'ファイル削除(読み取り専用も削除)
    file.Delete true
Next
```

```
'サブフォルダを列挙
```

```
For Each subfolder In objFolder.SubFolders
    'サブフォルダの中身を削除(再帰処理)
    Call DeleteFolder(strFolder & "¥" & subfolder.Name)
    'フォルダ内のファイルとフォルダがなくなったらフォルダを削除(読み取り専用も削除)
    If subfolder.Files.Count = 0 And subfolder.SubFolders.Count = 0 Then
        subfolder.Delete true
    End If
Next
```

```
End Sub
```

● PowerShell 版・・・kadai\_ps.bat を管理者として実行すること。datetime.bat は変更なし

### ■ kadai\_ps.bat

```
@echo off

rem 初期設定
set LOG_OUTPUT=C:¥temp¥kadai.log
set EXEC_PS1_FILE=C:¥temp¥kadai.ps1
set EXTRACT_FILE=C:¥temp¥kadai.zip
set DATETIME_BAT=C:¥temp¥datetime.bat
set START_MSG=***** ファイル解凍処理：開始 *****
set NORMAL_END_MSG=***** ファイル解凍処理：正常終了 *****
set ABNORMAL_END_MSG=***** ファイル解凍処理：異常終了
set NORMAL_RETURNCODE=0
set ABNORMAL_RETURNCODE=2

rem 処理開始メッセージ
call %DATETIME_BAT%
echo %START_MSG% %DATE% %TIME%
echo %START_MSG% %DATE% %TIME% > %LOG_OUTPUT%

rem 標準出力と標準エラー出力を両方ともログファイルに出力
rem タイトルロゴを出さない場合は、//nologo を指定
powershell Set-ExecutionPolicy RemoteSigned
powershell %EXEC_PS1_FILE% %EXTRACT_FILE% ;exit $LASTEXITCODE >>
%LOG_OUTPUT% 2>&1
set RC=%ERRORLEVEL%
powershell Set-ExecutionPolicy Restricted

rem 戻り値のチェック
if %RC%==%ABNORMAL_RETURNCODE% goto ERROR_END

rem 正常終了処理
call %DATETIME_BAT%
```

```
echo %NORMAL_END_MSG% %DATE% %TIME%  
echo %NORMAL_END_MSG% %DATE% %TIME% >> %LOG_OUTPUT%  
pause  
exit /b NORMAL_RETURNCODE
```

```
rem 異常終了処理  
:ERROR_END  
call %DATETIME_BAT%  
echo %ABNORMAL_END_MSG%(%RC%) ***** %DATE% %TIME%  
echo %ABNORMAL_END_MSG%(%RC%) ***** %DATE% %TIME% >> %LOG_OUTPUT%  
pause  
exit /b %RC%
```



## ■ kadai.ps1

```
# -----  
# ■ スクリプト内のすべての変数に対して明示的な宣言を強制する  
# -----  
  
set-psdebug -strict  
  
# -----  
# ■ 定数  
# -----  
  
set-variable -name START_MSG -value "==== ファイル解凍処理(PowerShell) : 開始 =====  
" -option constant  
set-variable -name END_MSG -value "==== ファイル解凍処理(PowerShell) : 終了 ====="  
-option constant  
set-variable -name NORMAL_END -value 0 -option constant  
set-variable -name ABNORMAL_END -value 2 -option constant  
set-variable -name ERROR_MSG_001 -value "引数の数が正しくありません。" -option constant  
set-variable -name ERROR_MSG_002 -value "引数が指定されていません。" -option constant  
set-variable -name ERROR_MSG_003 -value "引数で指定されたファイルが存在しません。"  
-option constant  
set-variable -name OUTPUT_PATH -value "C:¥temp¥sample¥" -option constant  
  
# -----  
# 関数名 : Extract . . . Zip ファイル解凍処理  
# 引数   : ZipFile . . . Zip ファイル名(フルパス)  
#       : ExtractTo . . . Zip ファイル解凍先フォルダ名(末尾に「¥」)  
# -----  
  
function Extract($ZipFile, $ExtractTo)  
{  
    $objShell = ""  
    $objFolder = ""
```

```

$FilesInZip = ""

# 解凍先フォルダが無い場合は作成
# 解凍先フォルダがある場合は、フォルダを削除して再作成

if (Test-Path $ExtractTo)
{
    # -Recurse オプションを付与することで、配下の要素を再帰的に削除することができる
    # 読み取り専用のファイルを強制的に削除したい場合には、-Force オプションを指定する
    Remove-Item $ExtractTo -Recurse -Force

    New-Item $ExtractTo -itemType Directory
}
else
{
    New-Item $ExtractTo -itemType Directory
}

# Zip ファイルを解凍して、全てのファイルを解凍先フォルダにコピー

$objShell = New-Object -ComObject shell.application
$objFolder = $objShell.NameSpace($ExtractTo)
$FilesInZip = $objShell.NameSpace($ZipFile)
$objFolder.CopyHere($FilesInZip.Items())

$ret = ""
return [void]$ret
}

# -----
# ■ 開始メッセージ出力
# -----

Write-Host $START_MSG

```

```
# -----  
# ■引数チェック  
# -----  
  
# 引数が1ではない場合、戻り値「2」でスクリプト終了  
# Write-Host でエラーメッセージを出力  
  
if (-not($args.Length -eq 1))  
{  
    Write-Host $ERROR_MSG_001  
    Write-Host $END_MSG  
    Write-Host $args.Length  
    exit $ABNORMAL_END  
}  
  
# 引数が空文字の場合、Write-Host でエラーメッセージを出力  
# 戻り値「2」でスクリプト終了。  
  
$strZipFile = $args[0]  
if ($strZipFile.Trim().Length -eq 0)  
{  
    Write-Host $ERROR_MSG_002  
    Write-Host $END_MSG  
    exit $ERROR_END  
}  
  
# 引数のファイルが存在しない場合は、Write-Host でエラーメッセージを出力  
# 戻り値「2」でスクリプト終了  
  
if (-not(Test-Path $strZipFile))  
{  
    Write-Host $ERROR_MSG_003  
    Write-Host $END_MSG
```

```
exit $ABNORMAL_END
```

```
}
```

```
#
```

---

```
# ■ C:¥temp¥sample フォルダは、無ければ作成
```

```
# 引数の zip ファイルを C:¥temp¥sample に解凍
```

```
#
```

---

```
Extract $strZipFile $OUTPUT_PATH
```

```
#
```

---

```
# ■ 終了メッセージ出力
```

```
# スクリプト終了。終了コード : 0
```

```
#
```

---

```
Write-Host $END_MSG
```

```
exit $NORMAL_END
```

## <VBScriptの参考情報>

VBScript 基礎文法最速マスター

<http://vbscript.g.hatena.ne.jp/cx20/20100131/1264906231>

スクリプトのインクルード

<http://sei.tee.jp/docs/program/hta/other/wsf.html>

バッチファイル(.bat)へ WSH(.wsf)のエラーコードの返し方について

<http://oshiete.goo.ne.jp/qa/2322637.html>

wsf(WSH)のデバッグ

<http://social.msdn.microsoft.com/forums/ja-JP/vwdexpressja/thread/eb13d7de-a6c1-4ed5-83d1-ccf159871525/>

VBS(WSF)サンプル

ユーザーをグループへ追加

<http://www.mitchy-world.jp/itmemo/vbs/03.htm>

WSFでインクルード

[http://seoi.net/wsh/base\\_include.shtml](http://seoi.net/wsh/base_include.shtml)

各種定義とソース分割

<http://winofsql.jp/VA003334/vbsguide051211173041.htm>

運用 Windows 管理者のための Windows Script Host 入門

第2回 WSH スクリプト・コーディングの第1歩

[http://www.atmarkit.co.jp/fwin2k/operation/wsh02/wsh02\\_01.html](http://www.atmarkit.co.jp/fwin2k/operation/wsh02/wsh02_01.html)

またまた言ってるそばから(汗)、.wsf と .vbs はVW2005Expressで編集できる

<http://d.hatena.ne.jp/katow/20081228/p7>

1.8 プログラムの再利用方法

<http://www.happy2-island.com/vbs/cafe02/capter00108.shtml>

4.VBScript のプログラムを作るときに使うツール

<http://www.happy2-island.com/vbs/cafe01/capter00400.shtml>

2.VBScript や WSH って何者? 何が得意?

<http://www.happy2-island.com/vbs/cafe01/capter00200.shtml>

VBScript でバッチ処理する時の選択肢: .VBS でやるか .WSF でやるか

<http://exvbs.seesaa.net/article/156736937.html>

Windows スクリプト ファイル (.wsf) を使用する

<http://msdn.microsoft.com/ja-jp/library/cc392507.aspx>

ADSI リファレンス

<http://msdn.microsoft.com/ja-jp/library/cc338173.aspx>

FTP 仮想サーバーを作成するための IIS サンプル ADSI スクリプト

<http://support.microsoft.com/kb/247364/ja>

ADSI の紹介

[http://www.anchor-systems.jp/anchor/ashp/netmon/samples/adsi\\_samples.html](http://www.anchor-systems.jp/anchor/ashp/netmon/samples/adsi_samples.html)

[ADSI プログラミング] VBScript のサンプル入手、生成方法について

<http://blogs.technet.com/b/jpilmblg/archive/2009/06/11/adsi-vbscript.aspx>

WMI サンプル集

[http://www.anchor-systems.jp/anchor/ashp/netmon/wmi\\_samples.html](http://www.anchor-systems.jp/anchor/ashp/netmon/wmi_samples.html)

WScript.CreateObject メソッドと CreateObject 関数についてです。

[http://detail.chiebukuro.yahoo.co.jp/qa/question\\_detail/q1341104333](http://detail.chiebukuro.yahoo.co.jp/qa/question_detail/q1341104333)

## <PowerShellの参考情報>

PowerShell 基礎文法最速マスター

<http://wincscript.jp/powershell/202>

Windows PowerShell

<http://ufcpp.net/study/powershell/index.html>

Windows PowerShell 入門

<http://codezine.jp/article/corner/10>

Windows PowerShell その1

<http://blog.goo.ne.jp/thaniwa/e/31c9de1562c12bc8f8178b3479f25118>

bat から ps1 ファイルを実行するときのあれこれ

<http://qiita.com/HiDARi/items/1d2595ff142883c3bac1>

PowerShell で再帰的なファイル削除の検証

<http://harikofu.blog.fc2.com/blog-entry-543.html>

PowerShell - 定数を使う

<http://www.itlab51.com/?p=5774>

PowerShell での文字列出力について考える

<http://tech.guitarrapc.com/entry/2014/01/17/073338>

PowerShell Tips ホーム

<http://hiros-dot.net/PowerShell/index.htm>

PowerShell でファイルやフォルダを操作する

<http://www.atmarkit.co.jp/fwin2k/win2ktips/1069psffolder/psffolder.html>

Zip ファイルを解凍する

<http://mtgpowershell.blogspot.jp/2011/02/zip.html>

PowerShell で Zip を回答する。(エクスプローラーの機能で)

<http://t100life.blog121.fc2.com/blog-entry-141.html>

PowerShell で Zip 圧縮、解凍処理に ZipFile クラスを試してみる(続き)

<http://tech.guitarrapc.com/entry/2013/10/08/040325>

ZIP ファイルを解凍する

<http://gallery.technet.microsoft.com/scriptcenter/ddbd1d0e-640b-4b80-8eab-ec06a636f006>

[Tips]DotNetZip ライブラリを使用して、Zip ファイルを解凍する

<http://blog.powershell-from.jp/?p=105>

ZIP ファイルダウンロード&展開をする PowerShell スクリプト

<http://qiita.com/kurukurupapa@github/items/99f739f0c0dd402e03ea>