VBScript und Objekte

Der Windows Scripting Host (WSH) stellt ein Objektmodell zur Verfügung, dass weitreichende Aufgaben ermöglicht. Suchen Sie in der Hilfe-Datei **`Script56.CHM`** mit dem Suchbegriff "***Objektmodell***" und wählen Sie in der Liste den Eintrag "Windows Script Host-Objektmodell". Dort sind alle Objekte in kurzer Form beschrieben.

VBScript kann jedoch noch viel mehr. Es können alle Programme sozusagen "ferngesteuert" werden, wenn diese ein Objektmodell veröffentlichen und nutzbar machen. Als Beispiel können hier die Programme der Office-Suite genannt werden.

Arbeitsfragen

1. Welches Objekt nutzen Sie, wenn eine Netzlaufwerkzuordnung hergestellt werden soll?
   _____
2. Welches Objekt ist geeignet eine Verknüpfung auf dem Desktop zu erstellen?
   _____
3. Durch welches Objekt lassen sich Umgebungsvariablen lesen?
   _____

Ein erstes Objekt

Im ersten VBScript soll gezeigt werden, dass VBScripte sogar Drag and Drop unterstützen.

Hierzu wird das Objekt **`WScript.Arguments`** verwendet.

```
'VBArgs.vbs
'Modifiziertes Beispiel aus der Hilfedatei Script56.CHM
Option Explicit

'Variablendeklaration
Dim objArgs
Dim i

'Herstellen eines Bezuges zum Objekt "WScript.Arguments"
Set objArgs = WScript.Arguments

'Das Objekt "WScript.Arguments" hat eine Methode mit dem
'Namen "Count", die die Anzahl der per Drag and Drop
'übergebenen Argumente enthält

For i = 0 to objArgs.Count - 1
   WScript.Echo i+1 & " Argument " &  objArgs(i)
Next
```

Mittels diesem Objekt lassen sich die bei Aufruf angegebenen Befehlszeilenargumente bzw. die per Drag and Drop auf das VBScript gezogenen Elemente wie Ordner oder Dateien bestimmen. Die Argumente sind sind als eine Liste in der Arguments-Eigenschaft des WScript-Objektes vorhanden. Die Anzahl dieser Listenelemente -d.h. die Anzahl der Argumente!- kann durch die Methode Count ermittelt werden.

Übung:

1. Geben Sie das obige Beispiel unter dem Namen `VBArgs.vbs` ein und machen Sie es ablauffähig (es sollte keine Fehlermeldung erscheinen, wenn Sie es in der GUI starten).
2. Testen Sie das Skript, in dem Sie einen Ordner oder eine Datei auf das Skript ziehen; es sollte dessen Name angezeigt werden. Ziehen Sie anschließend mehrere Dateien oder Ordner auf das Skript.
3. Ändern Sie das Skript so ab, dass zuerst die Anzahl der Argumente angezeigt wird.
4. Testen Sie das Skript auch in der Befehlszeile im Skriptordner durch Eingabe von `cscript vbargs.vbs test1 test2 test3`

Ein zweites Objekt

Sehen Sie sich nun das **WshEnvironment**-Objekt an. Dieses Objekt liefert eine Liste aller Umgebungsvariablen. Allerdings ist dieses Objekt ein "Unter-Objekt" des **WshShell**-Objektes wie Sie in dem Objektmodell sehen können. Auf das WshShell-Objekt kann nicht direkt zugegriffen werden! Solch ein Objekt muss zuerst erzeugt werden. Hierzu ist folgendes erforderlich:

```
'VBShell.vbs
'Demo für WScript.Shell Objekt
Option Explicit

'Erzeugen von Variablen, die den Objekt-Bezug aufnehmen
Dim wsh, EnvCollection

'Erzeugen des WshShell-Objektes und herstellen des Bezuges zum Objekt
Set wsh = CreateObject("WScript.Shell")

'Erzeugen eines Objektbezuges für eine Liste aller Umgebungsvariablen
'vom Typ "System". Man nennt solch eine Liste auch eine "Collection"
'Mögliche Typen sind "System", "Volatile", "User" und "Process"
Set EnvCollection = wsh.Environment("System")

'Objektvariable; nimmt jeweils ein Element der Collection auf
Dim Element
'Variable zur Ausgabe
Dim Liste

'Liste Element für Element zusammen stellen
for each Element in EnvCollection
         Liste=Liste & Element & vbcr
next
'Liste ausgeben
WScript.Echo Liste
```

In diesen Script taucht eine neue Form der for-Schleif auf: die `for...each...next`-Schleife. Diese Schleife ist hervorragend dazu geeignet, auf Objekte und deren Eigenschaften in Objektauflistungen (sogenannten "Collections") zuzugreifen. Sehen Sie sich mit mir das obige Beispiel näher an:

Die Objektvariable **`EnvCollection`** enthält die gesamte Liste aller Umgebungsvariablen.

Die Objektvariable mit dem Namen **`Element`** enthält nun im Rahmen der for-Schleife bei jedem Schleifendurchlauf den Wert **eines** Listenelementes. So kann die Liste Element für Element "abgearbeitet" werden und z.B. ein String zur Ausgabe mit der MessageBox-Anweisung zusammen gestellt werden (weiteres zur Schleife siehe auch im Buch **Scripting Host** Seite 54f.).

```
'Objektvariable für Auflistung der Umgebungsvarioablen
Dim EnvCollection
'Bezug zur Auflistung wird hergestellt
Set EnvCollection = wsh.Environment("System")

'Objektvariable für jeweils ein Element der Collection
Dim Element

'Liste Element für Element zusammen stellen
for each Element in EnvCollection
         Liste=Liste & Element & vbcr
next

MsgBox Liste
```

1. Erstellen Sie das Script-Programm `VBShell.vbs` und testen Sie es.
2. Ändern Sie das Script `VBArgs.vbs` so ab, dass die Ausgabe der Parameter mit einer MessageBox als Liste erfolgt und verwenden Sie hierzu die `for...each...next`-Schleife.