

Name:	Lehrgang: Betriebssysteme	Datum:
Arbeitsblatt Nr.	Windows Scripting Host: Wiederholungen	Seite 1 von 2

Wiederholungsstrukturen

In VB Skript existieren die gleichen Wiederholungsstrukturen wie in den klassischen Programmiersprachen, nämlich `for`- und `do`-Schleifen (kopf- und fußgesteuert).

Die For-Schleife

Anhand des nebenstehenden Beispiels soll die `for`-Schleife vorgestellt werden.

Die `for`-Schleife hat eine Laufvariable (hier `i`), der ein Startwert (hier `0`) zugewiesen wird. Die zwischen dem Schleifenkopf und der Anweisung `next` (gibt das Ende der Schleife an) enthaltenen Befehle werden ausgeführt, bis `i` den Wert von `Zahl` überschreitet (`i <= Zahl`)

```
'forDemo.vbs
'*****

Option Explicit
Dim i
Dim Zahl
dim Summe

Summe=0
Zahl=Inputbox("Summe aller ganzen Zahlen bis")

'Summenberechnung
for i=1 to Zahl
    Summe=Summe + i
next

msgbox "Die Summe lautet: " & Summe,,"Summenberechnung"
```

Übung:

1. Erstellen Sie das Programm `forDemo.vbs` und führen Sie es aus. Beseitigen Sie eventuelle Fehler! (Kontrolle: Summe aller Zahlen von 1 bis 10 ergibt 55)
2. Modifizieren Sie das Programm so, dass **nach** der Summenberechnung das Produkt der Zahlen (die Fakultät der Eingabe) berechnet wird. (Kontrolle: $10! = 3628800$)
3. Geben Sie die drei Beispielskripte 2-9.VBS bis 2-11.VBS aus dem Buch "Scripting Host" auf der Seite 31 ein und testen Sie diese. Sehen Sie sich die Möglichkeit zur Ausgabe mehrerer Werte in einer MessageBox mittels der beschriebenen Konstruktion auf Seite 32 an.
4. Modifizieren Sie das Skript aus Aufgabe 2 so, dass Summe und Produkt mit einer Messagebox in zwei separaten Zeilen angezeigt werden.
5. Informieren Sie sich über die VB-Funktion `IsNumeric()` und verändern Sie das Skript so, dass nur noch numerische Werte verarbeitet werden. Bei Eingabe von nichtnumerischen Werten soll eine geeignete Fehlermeldung erscheinen.

Die Do...Loop-Schleife

Wie auch in anderen Programmiersprachen, wird die Do...Loop-Schleife verwendet, wenn die Anzahl der Schleifendurchläufe im voraus unbekannt ist.

In VB-Skript gibt es prinzipiell vier Varianten, von denen allerdings zwei äquivalent sind.

do Befehle ... loop while Beding	do while Beding Befehle ... loop	do Befehle ... loop until Beding	do until Beding Befehle ... loop
--	--	--	--

Beding ist ein Ausdruck, der den Wert `true` oder `false` (wahr oder falsch) liefert.

Die Do...Loop-Schleife gibt es als kopfgesteuerte und als fussgesteuerte Variante. Bei der kopfgesteuerten Variante befindet sich der Bedingungsausdruck im Schleifenkopf (zweite und vierte Variante von links); bei der fussgesteuerten Variante befindet sich der Bedingungsausdruck im Schleifenfuss (erste und dritte Variante von links).

Bei der Laufbedingung kann nun noch wahlweise das Schlüsselwort **while** oder **until** verwendet werden. Beim Schlüsselwort `while` wird die Schleife ausgeführt, **solange** (=while) die Bedingung `true` ist. Wenn die Bedingung den Wert `false` liefert, bricht die Schleifenausführung ab. Beim Schlüsselwort `until` hingegen, wird die Schleife ausgeführt, **bis** (=until) die Bedingung `true` ist. Wenn die Bedingung den Wert `false` liefert, wird die Schleifen ausgeführt!

Mittels dem logischen **NOT** lässt sich also jederzeit eine while-Schleife in eine until-Schleife überführen; und umgekehrt! Es gibt keine feste Regel, wann man `while` bzw. `until` einsetzt. Besser ist zu überlegen, welche Logik im jeweiligen Programmkontext sinnvoller erscheint.

Beispiel: Es soll ein Datumswert per InputBox eingelesen werden und es soll sichergestellt werden, dass es sich auch um eine Eingabe handelt, die als Datumswert verstanden werden kann.

```
Option Explicit
Dim Datum

do
    Datum=InputBox("gültiges Datum eingeben!")
loop until IsDate(Datum)
MsgBox CDate(Datum)
```

Mit der Funktion `CDate()` wird eine gültige Datumseingabe in einen Datumswert konvertiert. dies hat den Vorteil, dass alle Datumseingaben (z.B. auch ein amerikanisches Format!) in gleichem Format (dd.mm.yyyy) vorliegen (hängt ab von der Ländereinstellung).

Übung:

1. Erstellen Sie das obige Beispielskript unter dem Namen `Datum.vbs` und führen Sie es aus. Ändern Sie das Skript so ab, dass das eingegebene Datum ohne `CDate()` ausgegeben wird. Verwenden Sie in beiden Fällen auch die Eingabe eines amerikanischen Datumswertes in der Form MM-DD-YYYY. Sehen Sie sich die Ergebnisse an! Was fällt ihnen auf?
2. Schreiben Sie ein VB-Skript namens `Dez2Bin.vbs`, dass eine positive ganze Dezimalzahl als Dualzahl ausgibt. Es soll überprüft werden, ob überhaupt eine Zahl eingegeben wurde. Falls ja, soll der eingegebene Wert, der noch einen String darstellt, in den Typ Integer konvertiert werden. Sollte eine negative Zahl eingegeben werden, soll eine geeignete Meldung erscheinen und der Absolutwert der Zahl verwendet werden. Informieren Sie sich hierzu über die Funktionen `IsNumeric()`, `CInt()`, `Abs()` und `StrReverse()` und die Operatoren `MOD` und `\` für spezielle Divisionen. (Legen Sie fest, dass alle Variablen deklariert werden müssen!)