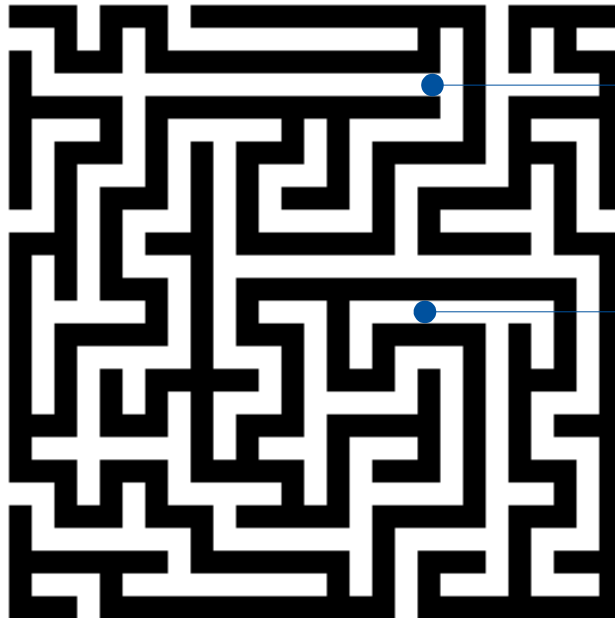


# Access 2010 – Programmierung Schleifen



Gehe nach links  
oder rechts

Gehe solange  
geradeaus...

## Beispiel für Schleifen

- In einem Formular gibt der Benutzer für den Farbanteil „Rot“ einen Unter- und Obergrenze ein. Die Werte liegen zwischen 0 und 255.
- Der rote Farbanteil wird mit blau und grün gemischt. Die gemischte Farbe wird mit Hilfe der Hintergrundfarbe eines Textfeldes angezeigt.
- Der angegebene Zahlenbereich für den Rot-Farbanteil wird in 20-Schritten durchlaufen. Die Farbe wird nach einer bestimmten Zeit gewechselt.

## Lösung

- An das Ereignis „Beim Klicken“ der Schaltfläche wird eine Prozedur angehängt.
- In dieser Prozedur wird zuerst abgefragt, ob die Ober- und Untergrenze für den Rot-Anteil zwischen 0 und 255 liegen.
- Falls ja, werden die Farben in Abhängigkeit einer bestimmten Schrittweite alle x Sekunden neu gemischt und angezeigt.

## Code-Schnipsel

```
For count = beginn To ende Step 20      ' Das Farbintervall wird durchlaufen
  If (count >= 0) And (count < 255) Then
    green = CInt(count / 2)

    blue = CInt(green * 3)
    If (blue > 255) Then
      blue = 255
    End If

    txtFarbe.BackColor = RGB(count, green, blue)
    lblFarbe.Caption = "RGB(" & count & ", " & green & ", " & blue & ")"
    Me.Refresh
  End If
Next
```

## Code-Schnipsel

```
For count = beginn To ende Step 20

    If (count >= 0) And (count < 255) Then
        ...

        start = Timer                ' Eieruhr starten

        Do While Timer < (start + pause)    ' ... und warten bis sie abgelaufen ist
            DoEvents
        Loop
    End If

Next
```

## Schleifen ...

- wiederholen Anweisungen in Abhängigkeit einer bestimmten Bedingung.
- nutzen einen Zahlenraum von bis.
- können geschachtelt werden.
- können vorzeitig beendet werden.

## Zählschleifen

```
For zählvariable = anfangswert To endwert Step schrittweite  
    Anweisungen  
Next
```

```
schritt = 1  
For count = intMin To intMax Step schritt  
    summe = summe + count  
Next
```

## Zählschleife ...

- läuft von ... bis.
- läuft eine genau festgelegte Anzahl von Durchläufen.
- hat den Schleifenkopf « For count = beginn To ende ». Für eine Zählvariable „count“ wird ein Anfangswert und ein Endwert festgelegt. Sobald der Endwert erreicht ist, wird die Schleife automatisch abgebrochen.
- endet mit dem Schlüsselwort « Next ». Dieses Schlüsselwort setzt die Zählvariable automatisch auf den nächsten Wert. Falls der Wert kleiner gleich dem Endwert ist, wird die Schleife nochmals durchlaufen. Andernfalls wird die Schleife abgebrochen.

## Schrittweite

- Das Intervall « count = beginn To ende » wird standardmäßig mit einer Schrittweite von 1 durchlaufen. Zum Beispiel für « beginn = 1 » hat „count“ im ersten Durchlauf den Wert 1, im zweiten Durchlauf 2 und so weiter. Die Schleife endet sobald die Variable „count“ einen Wert größer als „ende“ hat.
- Mit Hilfe von « For count = beginn To ende Step 20 » kann die Schrittweite angepasst werden. Zum Beispiel für « beginn = 1 » hat „count“ im ersten Durchlauf den Wert 1, im zweiten Durchlauf 21 und so weiter. Die Schleife endet sobald die Variable „count“ einen Wert größer als „ende“ hat.

## Hinweise zur Schrittweite

- Standardmäßig wird eine Schrittweite von 1 genutzt.
- Eine negative Schrittweite zählt, beginnend beim Startwert, runter.
- Eine positive Schrittweite zählt, beginnend beim Startwert, hoch.
- Falls der Datentyp der Schrittweite nicht gleich dem Datentyp der Zählvariablen ist, kann es zu Fehlern kommen.

## Nutzung einer negativen Schrittweite

```
Private Sub Form_Load()  
    Const maxAnzahl As Integer = 5  
    Const minAnzahl As Integer = 1  
    Dim aktuell As Integer  
    Dim count As Integer  
  
    aktuell = Year(Date)      ' Jahr des Systemdatums  
    IstJahre.AddItem aktuell  
  
    For count = minAnzahl To maxAnzahl  
        IstJahre.AddItem aktuell + count  
    Next  
End Sub
```

## Mehrfachauswahl bei einer Liste

```
Private Sub cmdAuswählen_Click()  
    Dim element As Variant  
  
    IstAuswahl.RowSourceType = "Value List"  
  
    For Each element In Me.IstJahre.ItemsSelected  
        IstAuswahl.AddItem IstJahre.Column(0, element)  
    Next  
  
End Sub
```

- Hinweis: Die Mehrfachauswahl einer Liste kann nur über die Registerkarte Andere im Eigenschaftenblatt eingestellt werden.

## Erläuterung

- Die Anweisungen werden für jedes Element in einem Feld (Array) oder Auflistung wiederholt.
- Für jedes Element in dem Feld oder der Auflistung, führe die dazugehörigen Anweisungen aus.
- Der nächste Schleifendurchlauf wird mit Hilfe des Schlüsselwortes « Next » gestartet.
- Die Schleife arbeitet nicht mit den Indizes, sondern durchläuft die Auflistung oder das Feld von vorn bis hinten.

## Aufbau des Schleifenkopfs

« For Each [Element in Feld] In [Feld, Auflistung]»

« For Each element In monatName »

- Der Schleifenkopf beginnt mit dem Schlüsselwort « For Each ».
- Dem Schlüsselwort folgt ein Platzhalter für jedes Element in dem Array oder der Auflistung. Falls ein Array genutzt wird, wird der Platzhalter als Variant gekennzeichnet.
- Dem Platzhalter folgt das Schlüsselwort « In », dem der Name der Auflistung oder dem Feld folgt. Das angegebenen Feld muss definiert sein.

## Do-Loop-Schleifen ...

- beginnen mit « Do ».
- beginnen mit « Loop » automatisiert einen neuen Schleifendurchlauf.
- können eine Abbruchbedingung haben, müssen aber nicht. Wenn keine Abbruchbedingung vorhanden ist oder diese nie erfüllt wird, läuft die Schleife endlos.

## Do-While-Schleifen

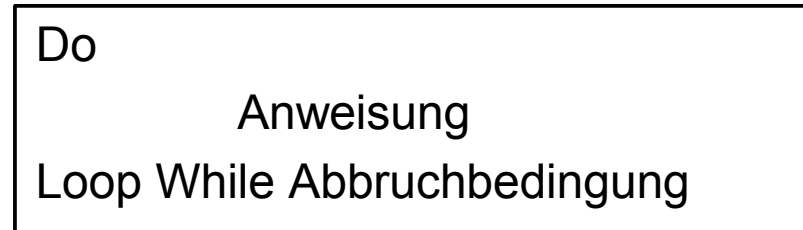
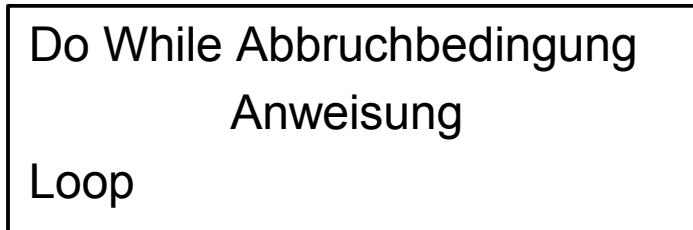
```
Do While Abbruchbedingung
        Anweisung
Loop
```

```
intMin = CInt(txtVon.Value)
intMax = CInt(txtBis.Value)
schritt = CInt(txtSchrittweite.Value)
summe = 0
count = intMin

Do While count <= intMax
    summe = summe + count

    count = count + schritt
Loop
```

## Abbruchbedingung « While » ...



- läuft solange die Bedingung erfüllt ist. Wenn die Bedingung nicht erfüllt ist, bricht die Schleife ab.
- kann im Kopf oder Fuß der Schleife stehen. Falls die Abbruchbedingung im Fuß steht, wird die Schleife mindestens einmal durchlaufen.

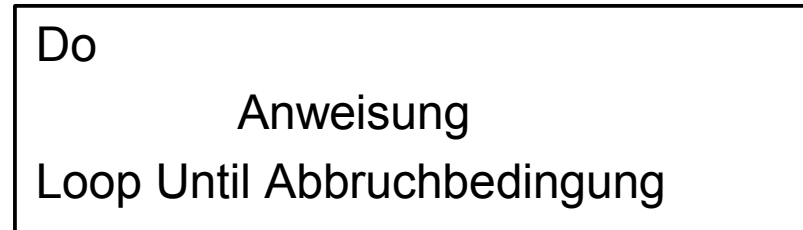
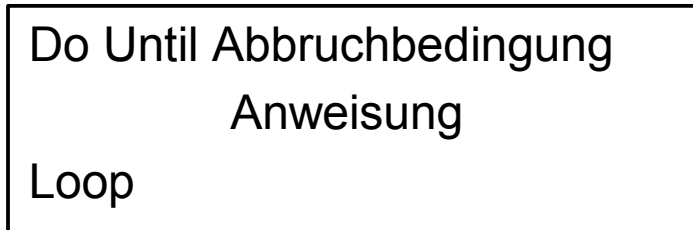
## Do-Until-Schleifen

```
Do Until Abbruchbedingung
    Anweisung
Loop
```

```
intMin = CInt(txtVon.Value)
intMax = CInt(txtBis.Value)
schritt = CInt(txtSchrittweite.Value)
summe = 0
count = intMin

Do Until count > intMax
    summe = summe + count
    count = count + schritt
Loop
```

## Abbruchbedingung « Until » ...



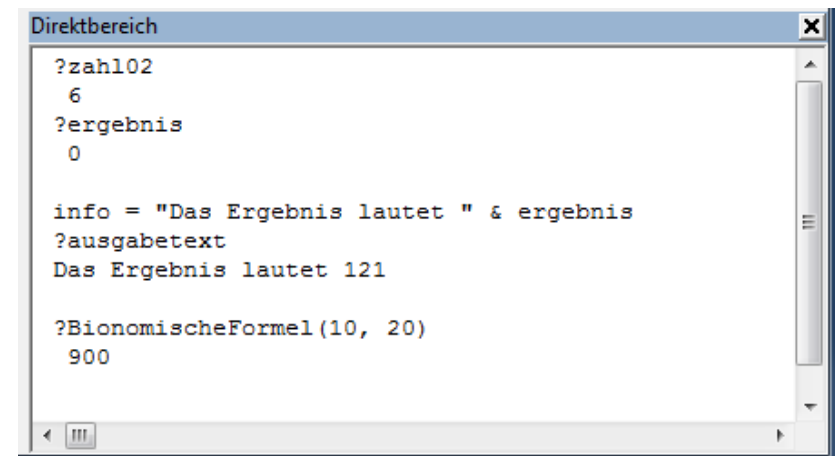
- läuft bis die Bedingung erfüllt ist. Wenn die Bedingung erfüllt ist, bricht die Schleife ab.
- kann im Kopf oder Fuß der Schleife stehen. Falls die Abbruchbedingung im Fuß steht, wird die Schleife mindestens einmal durchlaufen.

## Vorzeitiger Abbruch von Schleifen

- « Exit For » bricht die dazugehörige Zählschleife vorzeitig ab.
- « Exit Do » bricht die dazugehörige Do-Loop-Schleife vorzeitig ab.

## Direktfenster / -bereich zum Testen nutzen

- Welchen Wert eine Variable?
- Was passiert, wenn die Variable einen anderen Wert besitzt?
- Die Tests im Direktfenster werden immer in einem bestimmten Kontext gestartet.



```
Direktbereich
?zahl102
6
?ergebnis
0

info = "Das Ergebnis lautet " & ergebnis
?ausgabertext
Das Ergebnis lautet 121

?BionomischeFormel(10, 20)
900
```

## Voraussetzung

- Das Programm wird im Einzelschritt-Modus durchlaufen.
- Das Direktfenster wird mit Hilfe des Menüs *Ansicht – Direktfenster* eingeblendet.
- Die zu testenden Variablen sind in der zu durchlaufenden Prozedur definiert.

## Eingabe in das Direktfenster

- Anweisungen können manuell eingegeben oder aus dem Code-Bereich kopiert werden.
- Pro Zeile wird eine Anweisung in das Direktfenster eingegeben.
- Jede Zeile wird mit <RETURN> abgeschlossen. Die Anweisung wird automatisch ausgeführt.

## Test-Möglichkeiten im Direktfenster

- « variable = ausdruck ». Der Variablen wird ein neuer Wert zugewiesen. Der Code wird nicht verändert, nur der Wert der Variablen für das Debuggen des Codes.
- « ?variable ». Der Wert einer Variablen wird angezeigt.

## Anweisungen aus dem Direktfenster löschen

- Mit Hilfe der Maus werden Anweisungen im Direktfenster markiert. Mit Hilfe der Tastenkombination <STRG>+<A> wird das Direktfenster vollständig markiert.
- Mit Hilfe von <ENTF> wird der markierte Text aus dem Direktfenster gelöscht.

## Test-Anweisung im Codebereich

- « `Debug.Print ausdruck` ». Der Wert des berechneten Ausdrucks oder Variablen wird im Direktfenster angezeigt.
- « `Debug.Assert ausdruck` » unterbricht die Ausführung, sobald der Ausdruck oder die Variable den Wert `false` (falsch) hat.