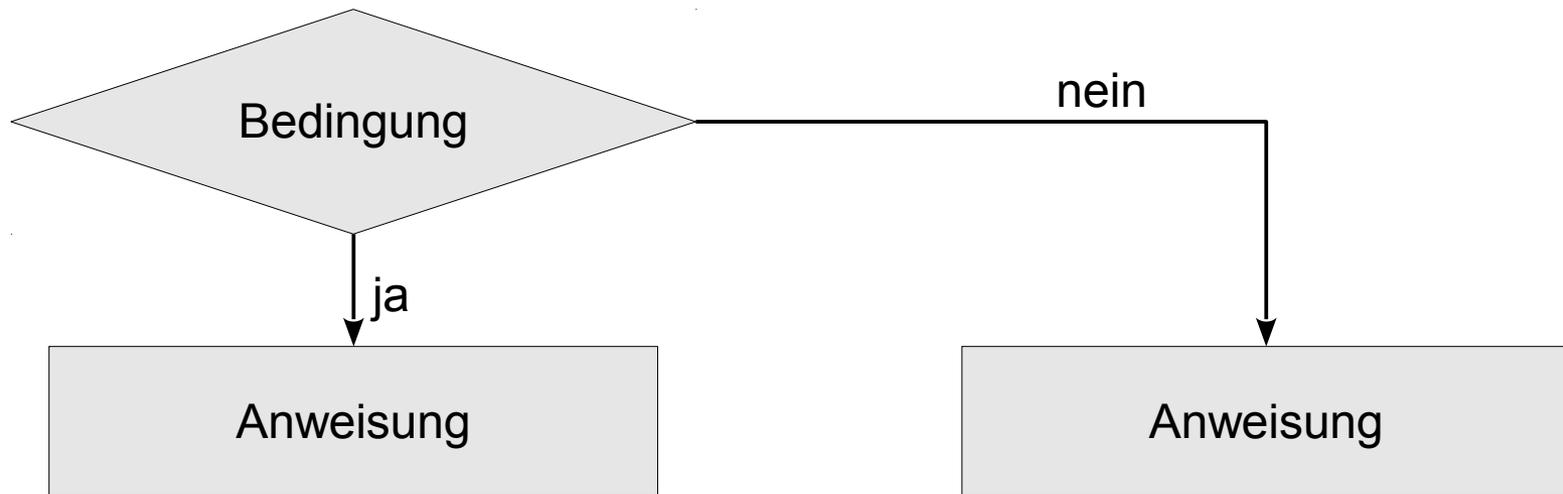


# V(isual) B(asic for) A(pplication) Bedingte Anweisungen



# Bedingte Anweisung

```
Dim temperatur As Double
```

```
temperatur = 0.6
```

```
If (temperatur > 0#) Then  
    Debug.Print "Plus-Temperatur"
```

```
ElseIf (temperatur < 0#) Then  
    Debug.Print "Minus-Temperatur"
```

```
Else  
    Debug.Print "Null-Punkt"
```

```
End If
```

# Wenn-Dann-Anweisungen

- Wenn die Temperatur einen bestimmten Wert erreicht hat, dann ...
- Wenn das Datenfeld A den Wert x gespeichert hat, dann ...
- Wenn der Kontostand dem Dispo übersteigt, dann ...
- Wenn die Strecke A doppelt so lang ist wie Strecke B, dann ...
- Wenn die Warenmenge eine Mindestmenge unterschreitet, dann ...

## if-Anweisung

```
Dim temperatur As Double
```

```
    temperatur = 0.6
```

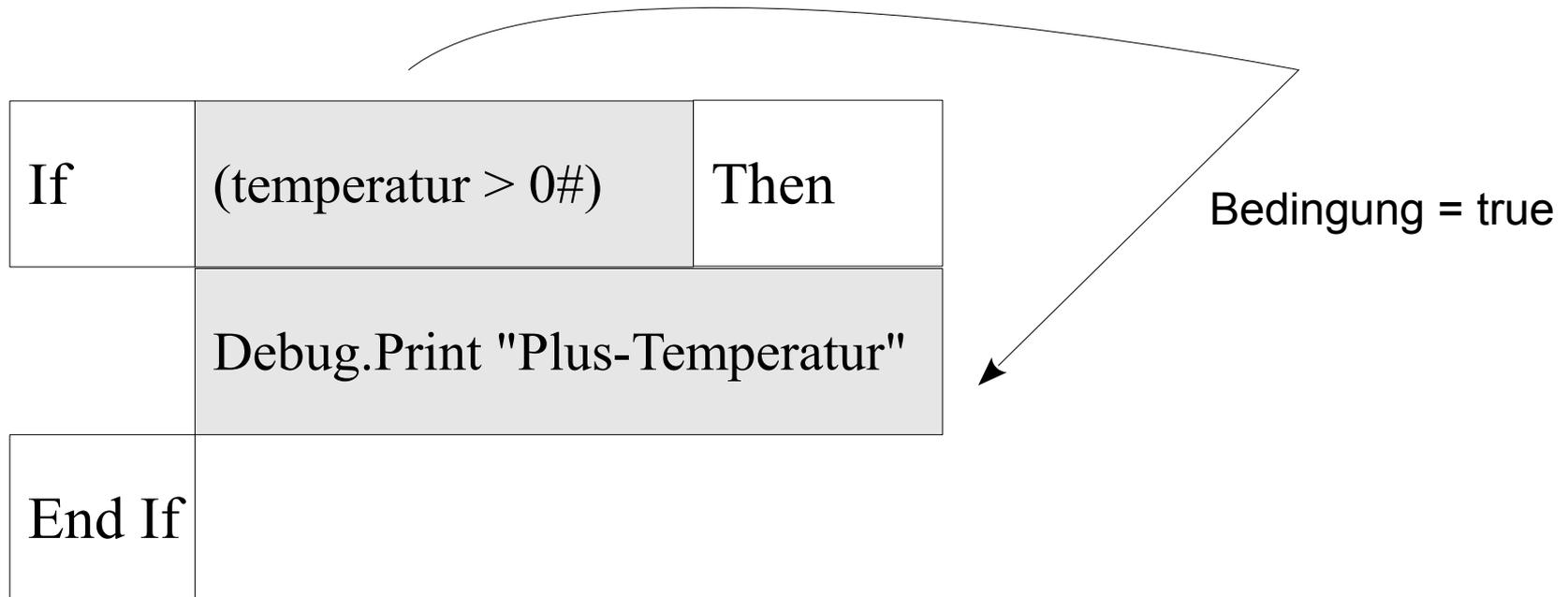
```
    If (temperatur > 0#) Then
```

```
        Debug.Print "Plus-Temperatur"
```

```
    End If
```

- Wenn (if) die Bedingung zutrifft, dann (then) führe die Anweisungen aus ...
- Wenn die Bedingung wahr ist, werden die darunter stehenden Anweisungen ausgeführt.

# Arbeitsweise



# Aufbau

If	[Bedingung]	Then
	[Anweisung]	
	[Anweisung]	
End If		

## Kopf der if-Anweisung

If (temperatur > 0#) Then

- Beginn mit dem Schlüsselwort If.
- Dem Schlüsselwort folgt die Bedingung. Die Bedingung kann, muss aber nicht geklammert werden.
- Der Kopf der Anweisung endet mit dem Schlüsselwort Then.

## Rumpf der if-Anweisung

```
If (temperatur > 0#) Then  
    Debug.Print "Plus-Temperatur"
```

- Wenn die Bedingung wahr ist, werden die Anweisungen die, dem Schlüsselwort Then folgen, ausgeführt.
- Die Anweisungen sollten entsprechend der if-Anweisung eingerückt werden.

## Ende der if-Anweisung

```
If (temperatur > 0#) Then  
    Debug.Print "Plus-Temperatur"  
End if
```

- Die if-Anweisung endet mit den Schlüsselwörtern End If.

# Bedingungen

- Ausdrücke, die einen boolschen Wert zurückgeben.
- Vergleich von Werten. Beantwortung von Fragen wie zum Beispiel „Ist der Messwert größer als 0“.

## Beispiel

```
temperatur = 0.6
```

```
If (temperatur > 0#) Then  
    Debug.Print "Plus-Temperatur"
```

- Mit Hilfe von Vergleichsoperatoren werden zwei Werte verglichen. In diesem Beispiel wird der Wert in der Variablen temperatur mit Null verglichen.
- Vergleichsoperatoren können aus ein oder zwei Zeichen bestehen. Zwischen den Zeichen darf kein Leerzeichen stehen.

# Vergleichsoperatoren

Ist kleiner

5	<	2	=	var false
---	---	---	---	--------------

Ist kleiner gleich

5	<=	2	=	var false
---	----	---	---	--------------

Ist größer

5	>	2	=	var true
---	---	---	---	-------------

Ist größer gleich

5	>=	2	=	var true
---	----	---	---	-------------

# Vergleichsoperatoren „Ist gleich“ und „Ist ungleich“

Ist gleich

5	=	2	=	var
				false

Ist nicht gleich

5	<>	2	=	var
				true

## Beispiel: Ist gleich

```
If (rs.Fields("Auslaufartikel").Value = True) Then
```

```
    strProdukt = rs.Fields("Produktcode").Value()
```

```
    strProdukt = strProdukt & ": " & rs.Fields("Artikelname").Value()
```

```
    Debug.Print strProdukt & " ist ein Auslaufartikel."
```

```
End If
```

- Das Feld in der Tabelle ist vom Datentyp Boolean. Ist die Aussage „Ist das Produkt ein Auslaufartikel?“ wahr?
- Ist das Kontrollkästchen aktiv?
- Entspricht die Variable exakt dem angegebenen Wert?  
Gleitkommazahlen sollten nicht auf Gleichheit überprüft werden.

## Beispiel: Ist ungleich

```
If (rs.Fields("Auslaufartikel").Value <> True) Then
```

```
    strProdukt = rs.Fields("Produktcode").Value()
```

```
    strProdukt = strProdukt & ": " & rs.Fields("Artikelname").Value()
```

```
    Debug.Print strProdukt & " ist ein Auslaufartikel."
```

```
End If
```

- Das Feld in der Tabelle ist vom Datentyp Boolean. Ist die Aussage „Ist das Produkt ein Auslaufartikel?“ falsch?
- Ist das Kontrollkästchen deaktiviert?
- Entspricht die Variable nicht dem angegebenen Wert?

## Leere Zeichenfolge

```
Dim wort As String
```

```
If (wort = "") Then  
    wort = "abcd"
```

```
End If
```

- Der String hat einen definierten Wert.
- Wenn die Bedingung wahr ist, enthält die Variable `wort` eine leere Zeichenfolge.

## Undefiniertes Datenfeld

```
rs.MoveFirst
```

```
Do While Not (rs.EOF)
```

```
    If IsNull(rs.Fields("Versanddatum").Value) Then
```

```
        bestellnummer = rs.Fields("Bestell-Nr").Value()
```

```
        Debug.Print "Keine Versendung"
```

```
    End If
```

```
    rs.MoveNext
```

```
Loop
```

## Erläuterung

- Der Funktion `IsNull()` wird das zu untersuchende Element übergeben.
- Wenn das Datenfeld einen undefinierten Inhalt enthält, ist die Bedingung wahr.

## Is Nothing

```
If dbs Is Nothing Then  
    Debug.Print "Kein Verweis vorhanden"  
    Exit Sub  
End If
```

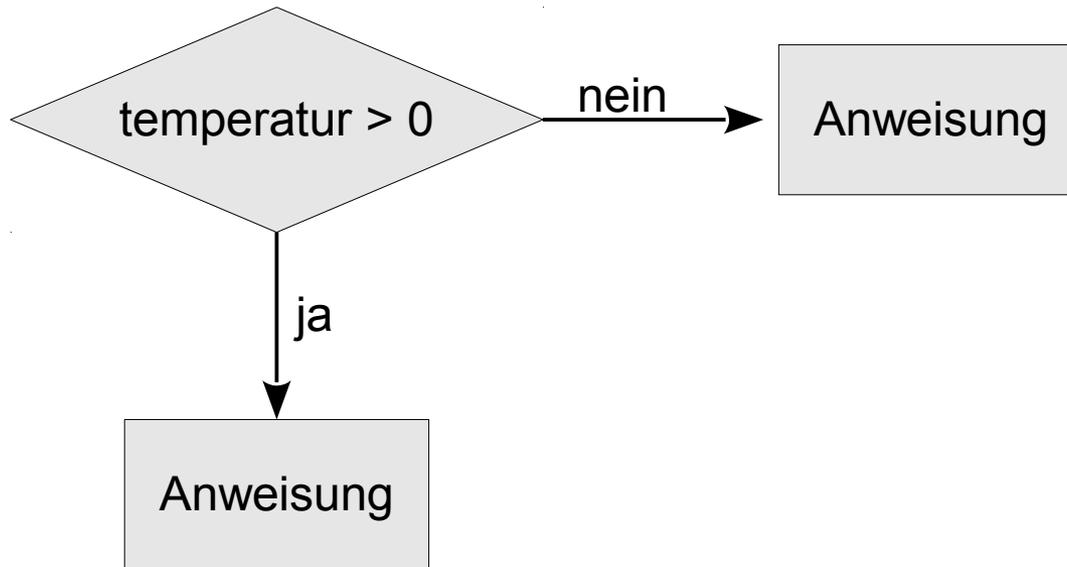
- Ist die Objekt-Variable leer?
- Wenn kein gültiger Verweis vorhanden ist, wird die Prozedur mit Exit Sub abgebrochen.

## ... andernfalls

```
If (temperatur > 0#) Then
    Debug.Print "Plus-Temperatur"
Else
    Debug.Print "Null-Punkt"
End If
```

- Andernfalls (Else) führe ... aus.
- Beschreibung des Standardfalls.
- Else-Anweisungen beziehen sich immer auf eine if-Anweisung.
- Die else-Anweisung und die dazugehörige Anweisung haben die gleiche Einrücktiefe.
- Die else-Anweisung ist optional.

# Grafische Darstellung



# Fallunterscheidung

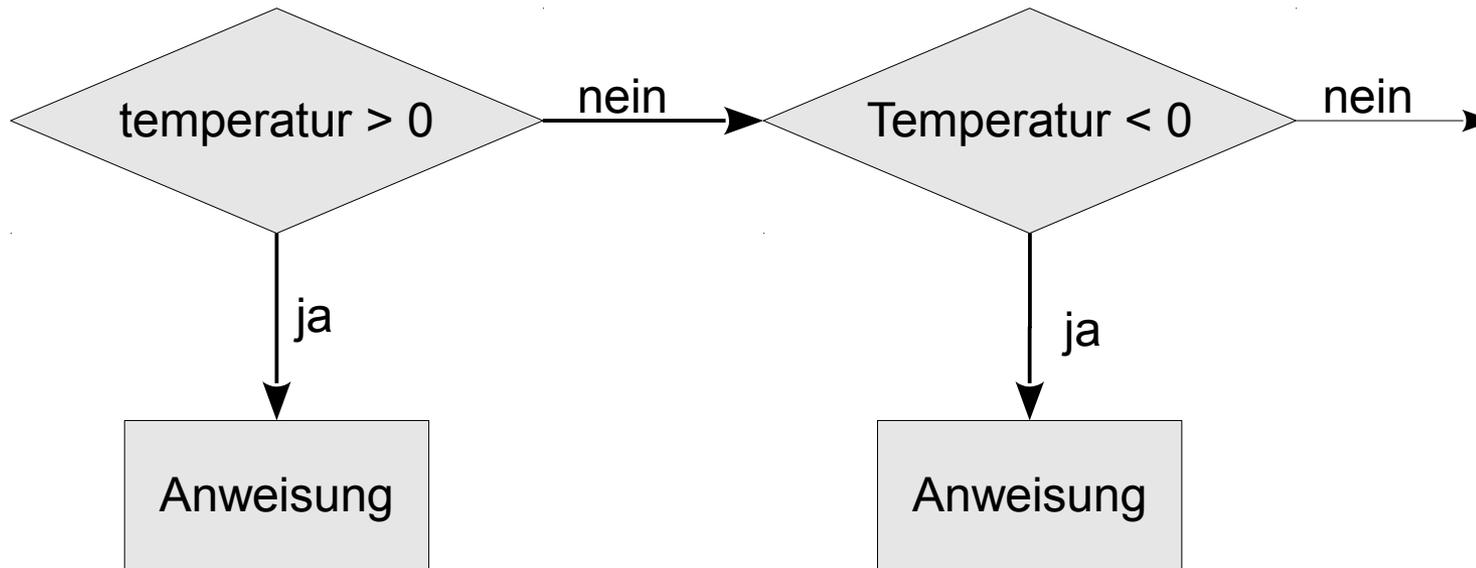
```
If (temperatur > 0#) Then
    Debug.Print "Plus-Temperatur"

ElseIf (temperatur < 0#) Then
    Debug.Print "Minus-Temperatur"

Else
    Debug.Print "Null-Punkt"

End If
```

# Grafische Darstellung



## Erläuterung

- Im ersten Schritt wird die Bedingung der if-Anweisung überprüft.
- Falls diese nicht zutrifft, wird die erste else if-Anweisung überprüft. Wenn die Bedingung zutrifft, wird der dazugehörige Anweisungsblock ausgeführt. Alle nachfolgenden Bedingungen sowie die else-Anweisung werden ignoriert.
- Falls diese nicht zutrifft, wird die nächste else if-Anweisung überprüft.
- Falls keine der Bedingungen zutrifft, wird, falls vorhanden, die else-Anweisung ausgeführt.

## Bedingte Anweisungen verschachteln

```
If IsNull(rs.Fields("Versanddatum").Value) Then  
  
    maxVersand = DateAdd("m", 3, rs.Fields("Bestelldatum").Value)  
  
    If (Date > maxVersand) Then  
        bestellnummer = rs.Fields("Bestell-Nr").Value()  
        Debug.Print "Keine Versendung"  
    End If  
  
End If
```

# Logische Operatoren

- Verknüpfungen von Ausdrücken, die einen booleschen Wert zurückliefern.
- Und-Verknüpfung. Beide Bedingungen müssen erfüllt sein.
- Oder-Verknüpfung. Eine der beiden Bedingungen müssen erfüllt sein.
- Nicht. Der Wert der Bedingung wird negiert.

# Und-Verknüpfung

a		b		
true	and	true	=	true
false	and	false	=	false
true	and	false	=	false
false	and	true	=	false

## Beispiel „und“

```
If (temperatur >= 36.9) And (temperatur <= 37.4) Then  
    Debug.Print "Erhöhte Temperatur"
```

```
ElseIf (temperatur >= 37.5) And (temperatur <= 39.4) Then  
    Debug.Print "Fieber"
```

```
ElseIf (temperatur >= 39.5) And (temperatur <= 40.5) Then  
    Debug.Print "Hohes Fieber"
```

```
ElseIf (temperatur >= 40.5) Then  
    Debug.Print "Sehr hohes Fieber"
```

```
Else  
    Debug.Print "Normaltemperatur"  
End If
```

## Hinweise

```
If (temperatur >= 36.9) And (temperatur <= 37.4) Then  
    Debug.Print "Erhöhte Temperatur"
```

```
ElseIf (temperatur >= 37.5) And (temperatur <= 39.4) Then  
    Debug.Print "Fieber"
```

```
End If
```

- Wenn die linke Bedingung wahr ist, wird die rechte Bedingung überprüft.
- Werte zwischen 37.4 und 37.5 werden nicht abgefangen.

# Oder-Verknüpfung

a		b		
true	or	true	=	true
false	or	false	=	false
true	or	false	=	true
false	or	true	=	true

## Beispiel „oder“

```
If (monat = 12) Or (monat = 1) Or (monat = 2) Then
    Debug.Print "Winter"

ElseIf (monat = 3) Or (monat = 4) Or (monat = 5) Then
    Debug.Print "Frühjahr"
ElseIf (monat = 6) Or (monat = 7) Or (monat = 8) Then
    Debug.Print "Sommer"
ElseIf (monat = 9) Or (monat = 10) Or (monat = 11) Then
    Debug.Print "Herbst"

Else
    Debug.Print "Keine korrekte Monatsangabe"
End If
```

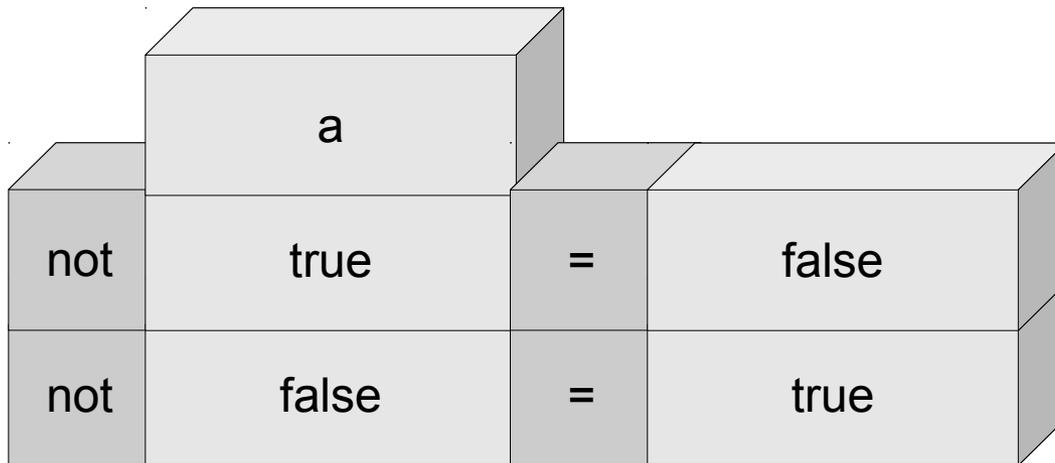
## Hinweise

```
If (monat = 12) Or (monat = 1) Or (monat = 2) Then  
    Debug.Print "Winter"
```

```
ElseIf (monat = 3) Or (monat = 4) Or (monat = 5) Then  
    Debug.Print "Frühjahr"
```

- Wenn die linke Bedingung falsch ist, wird die rechte Bedingung überprüft.

# Nicht-Verknüpfung

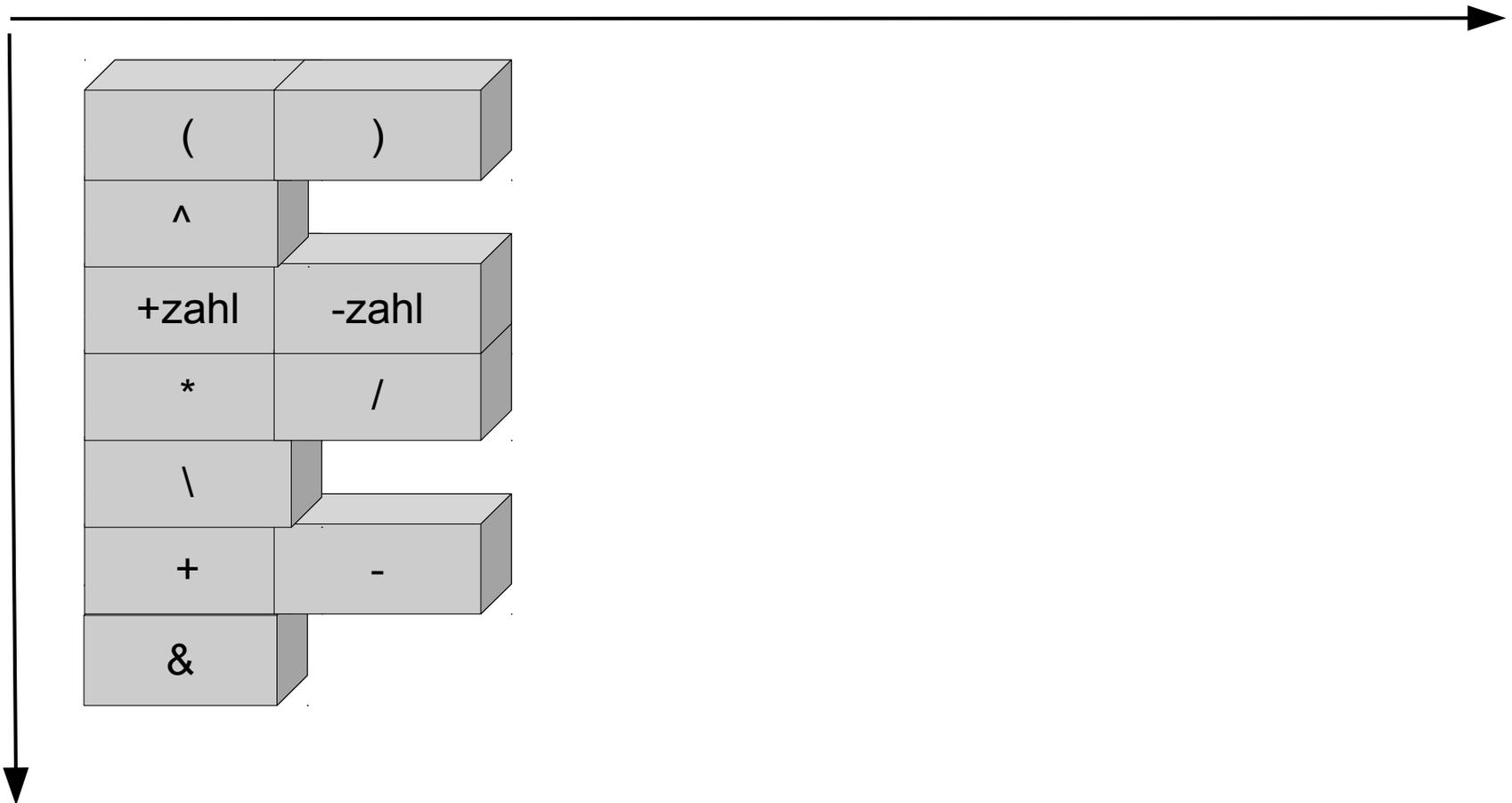


## Not Is Nothing

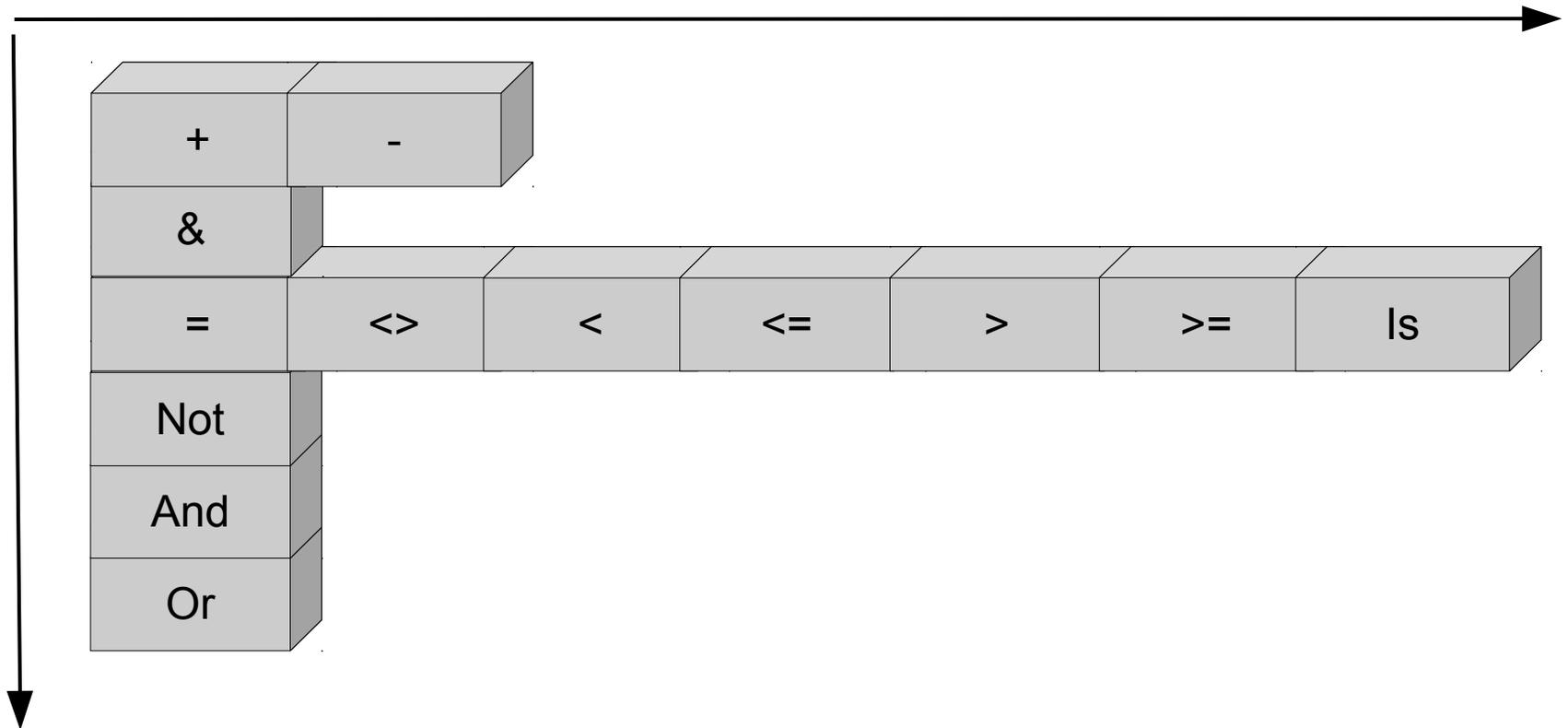
```
If Not (rs Is Nothing) Then  
    rs.MoveLast  
End If
```

- Der Operator Not negiert die Bedingung.
- Durch den Operator wird True zu False und umgekehrt.

# Rangfolge



# Rangfolge



## Vergleiche von Strings

```
If (rs.Fields("Kategorie").value() Like "Getränke") Then
    strProdukt = rs.Fields("Produktcode").Value()
    strProdukt = strProdukt & ": " & rs.Fields("Artikelname").Value()

End If
```

- Mit Hilfe des Schlüsselwortes LIKE können Strings mit einem Textmuster verglichen werden.
- Textmuster beginnen und enden mit den Anführungsstrichen.

## Platzhalter „Sternchen“

```
If (rs.Fields("Produktcode") Like "NWTCO-*") Then  
    strProdukt = rs.Fields("Produktcode").Value()  
    strProdukt = strProdukt & ": " & rs.Fields("Artikelname").Value()  
End If
```

- Das Sternchen steht für eine beliebige Anzahl von Zeichen oder keines.
- Das Sternchen kann an jeder beliebigen Position im Textmuster stehen.

## Platzhalter „Fragezeichen“

```
If (rs.Fields("Produktcode") Like "NWT?-*") Then  
    strProdukt = rs.Fields("Produktcode").Value()  
    strProdukt = strProdukt & ": " & rs.Fields("Artikelname").Value()  
End If
```

- Das Fragezeichen steht für exakt ein alphanumerisches oder numerisches Zeichen.
- Das Fragezeichen kann an jeder beliebigen Position im Textmuster stehen.

## Platzhalter „Liste von Zeichen“

```
If (strNachname Like "[AB]*") Then  
    Debug.Print strNachname  
  
End If
```

- Mit Hilfe der eckigen Klammern wird eine Liste von Zeichen zusammengefasst.
- Die Zeichen dürfen an dieser Position vorkommen.
- In diesem Beispiel darf der Nachname mit dem Buchstaben A oder B beginnen.

## Negieren des Platzhalter „Liste von Zeichen“

```
If (strNachname Like "[!AB]*") Then  
    Debug.Print strNachname  
  
End If
```

- Mit Hilfe der eckigen Klammern wird eine Liste von Zeichen zusammengefasst.
- Das Aufrufezeichen negiert die Liste. Die Zeichen dürfen an dieser Position nicht vorkommen.
- In diesem Beispiel darf der Nachname nicht mit dem Buchstaben A oder B beginnen.

## „Wähle aus“

```
Select Case rs.Fields("Status-Nr").Value
  Case 0
    Debug.Print "neu angelegt."
  Case 1
    Debug.Print "in Rechnung gestellt."
  Case 2
    Debug.Print "versandt."
  Case 3
    Debug.Print "geschlossen."
  Case Else
    Debug.Print "ohne Statusangabe."

End Select
```

# Aufbau

Select	Case	[variable]	
	Case	Bedingung	:
		Anweisung	
	Case Else	Bedingung	:
		Anweisung	
End Select			

## Kopf der Select Case - Anweisung

Select	Case	intKategorie
Select	Case	[variable]

- Den Schlüsselwörtern Select Case folgt der Name der zu untersuchenden Variable.
- Variablen vom Typ „String“, „Zahl“ oder „Datums-/Zeitwerte“ können untersucht werden.

## Beispiel: Überprüfung von Datumsangaben

```
Dim aktuellDatum As Date
aktuellDatum = #12/24/2017#

Select Case aktuellDatum
    Case #12/24/2017#
        Debug.Print "Heiligabend"
    Case #12/25/2017#
        Debug.Print "1. Weihnachtstag"
    Case #12/26/2017#
        Debug.Print "2. Weihnachtstag"
    Case #12/31/2017#
        Debug.Print "Silvester"
    Case #1/1/2018#
        Debug.Print "Neujahr"
End Select
```

## Fälle

Case	#12/24/2017#
Case	[wert]

- Den Schlüsselwörtern Case folgt ein Wert.
- Jeder Fall entspricht der Bedingung (`[variable] = [wert]`).
- Die einzelnen Fälle werden von oben nach unten durchlaufen.
- Die Werte der verschiedenen Fälle sind vom gleichen Datentyp der zu überprüfenden Variable oder können in diesen konvertiert werden.

## Standardfall

- Die Schlüsselworte `Case Else` beschreiben den Standardfall.
- Der Standardfall wird ausgeführt, wenn keine anderen Fälle zutreffen.
- Der Fall `Case Else` ist optional.

## Beispiel: Nutzung einer Auflistungen

```
Dim land As String  
land = "Polen"  
  
Select Case land  
    Case "Polen", "Dänemark", "Niederlande"  
        Debug.Print "Europa"  
  
    Case "Indien", "China"  
        Debug.Print "Asien"  
  
End Select
```

## Erläuterung

Case	"Indien"	,	"China"
Case	[wert]	,	[wert]

- Den Schlüsselwörtern Case folgt eine Auflistung von Werten.
- Die Werte in der Auflistung werden durch ein Kommata getrennt.
- Jeder Fall entspricht der Bedingung ([variable] = [wert]) OR ([variable] = [wert]).

## Beispiel: Nutzung von Wertebereichen

```
Select Case monat
  Case 3 To 5
    Debug.Print "Frühling"

  Case 6 To 8
    Debug.Print "Sommer"

  Case 9 To 11
    Debug.Print "Herbst"

End Select
```

## Erläuterung

Case	4	To	5
Case	[min]	To	[max]

- Den Schlüsselwörtern Case folgt ein Wertebereich.
- Der Wertebereich wird durch einen unteren und oberen Wert begrenzt.
- Jeder Fall entspricht der Bedingung  $([variable] \geq [wert]) \text{ AND } ([variable] \leq [wert])$ .

## Beispiel: Nutzung von Vergleichsoperatoren

```
Select Case temperatur
  Case 36.9 To 37.4
    Debug.Print "Erhöhte Temperatur"

  Case Is > 37.4 And temperatur <= 39.4
    Debug.Print "Fieber"

  Case Is > 39.4 And temperatur <= 40.5
    Debug.Print "Hohes Fieber"

  Case Is > 40.5
    Debug.Print "Sehr hohes Fieber"
End Select
```

## Erläuterung

Case	Is	>	40.5
Case	Is	[operator]	[wert]

- Den Schlüsselwörtern Case folgt eine Bedingung.
- Das Schlüsselwort Is ist ein Platzhalter für die zu überprüfende Variable.

# Verknüpfung von Bedingungen

```
Select Case temperatur  
  Case Is > 37.4 And temperatur <= 39.4  
    Debug.Print "Fieber"  
End Select
```

- Die erste Bedingung beginnt mit Case Is. Dem Schlüsselwort Is folgt der Vergleichsoperator und der Vergleichswert.
- Bedingungen können mit AND oder OR verknüpft werden.
- In der zweiten Bedingung muss der Name der zu überprüfenden Variablen genutzt werden.

## Hinweis

```
Select Case temperatur
```

```
Case Is >= 36.9 And temperatur <= 37.4  
  Debug.Print "Erhöhte Temperatur"
```

```
End Select
```

- Der oben dargestellte Fall ist immer wahr.
- Die Fallunterscheidung entspricht einem Zahlenraum. Ein Zahlenraum wird folgendermaßen dargestellt: 36.9 To 37.4.

## Beispiel: Nutzung von Vergleichsoperatoren

```
Select Case temperatur
  Case Is >= 36.9 And (temperatur <= 37.4)
    Debug.Print "Erhöhte Temperatur"

  Case Is > 37.4 And temperatur <= 39.4
    Debug.Print "Fieber"

  Case Is > 39.4 And temperatur <= 40.5
    Debug.Print "Hohes Fieber"

  Case Is > 40.5
    Debug.Print "Sehr hohes Fieber"
End Select
```