

# REPEAT

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiele

Verwandte Statements: FOR | ESCAPE

Gehört zur Funktionsgruppe: *Schleifenverarbeitung*

---

## Funktion

Mit dem Statement REPEAT können Sie eine Verarbeitungsschleife initiieren.

## Syntax-Beschreibung

Zwei unterschiedliche Strukturen sind bei diesem Statement möglich:

- Syntax 1 — Statements werden ein- oder mehrmals ausgeführt
- Syntax 2 — Statements werden überhaupt nicht oder mehrmals ausgeführt

Wann die Bedingung ausgewertet wird, ist abhängig davon, ob Sie sie an den Anfang oder das Ende der logischen Bedingung stellen.

Weitere Informationen zu logischen Bedingungen, siehe den Abschnitt *Logische Bedingungen* im *Leitfaden zur Programmierung*.

Eine Erläuterung der in den Syntax-Diagrammen benutzten Symbole siehe *Syntax-Symbole*.

Syntax 1:

<b>REPEAT</b>	
<i>statement</i>	''' [ { UNTIL } logical-condition ] [ WHILE ]
<b>END-REPEAT</b>	(nur im Structured Mode)
[ LOOP ]	(nur im Reporting Mode)

Syntax 2:

<b>REPEAT</b>	
$\left[ \left\{ \begin{array}{l} \text{UNTIL} \\ \text{WHILE} \end{array} \right\} \right]$	<i>logical-condition</i>
	$] \text{ statement...}$
<b>END-REPEAT</b> ( <i>nur im Structured Mode</i> )	
[ <b>LOOP</b> ] ( <i>nur im Reporting Mode</i> )	

Syntax-Element-Beschreibung:

<b>UNTIL</b>	Die Schleife wird so oft ausgeführt, bis die logische Bedingung erfüllt ist.
<b>WHILE</b>	Die Schleife wird solange ausgeführt, wie die logische Bedingung erfüllt ist.
<i>logical-condition</i>	Wenn Sie eine logische Bedingung angeben, bestimmt die Bedingung, wann die Ausführung der Schleife beendet werden soll.  Wenn Sie keine logische Bedingung angeben, müssen Sie die Schleife mit einem ESCAPE-, STOP- oder TERMINATE-Statement beenden, das in der Schleife angegeben ist.
<b>END-REPEAT</b>	Das für Natural reservierte Wort END-REPEAT muss zum Beenden des REPEAT-Statements benutzt werden.

## Beispiele

- Beispiel 1 — REPEAT-Statement
- Beispiel 2 — REPEAT-Statement mit Optionen WHILE und UNTIL

### Beispiel 1 — REPEAT-Statement

```

** Example 'RPTEX1S': REPEAT (structured mode)
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 PERSONNEL-ID
  2 NAME
*
1 #PERS-NR (A8)
END-DEFINE
*
REPEAT
  INPUT 'ENTER A PERSONNEL NUMBER:' #PERS-NR
  IF #PERS-NR = ' '
    ESCAPE BOTTOM
  END-IF
/*
  FIND EMPLOY-VIEW WITH PERSONNEL-ID = #PERS-NR
  IF NO RECORD FOUND
    REINPUT 'NO RECORD FOUND'
  END-NOREC
  DISPLAY NOTITLE NAME

```

```

    END-FIND
END-REPEAT
*
END

```

Ausgabe des Programms RPTEX1S:

ENTER A PERSONNEL NUMBER: **11500304**

Nach Eingabe und Bestätigung der Personalnummer:

```

        NAME
-----
KLUGE

```

Äquivalentes Reporting-Mode-Beispiel: RPTEX1R.

## Beispiel 2 — REPEAT-Statement mit Optionen WHILE und UNTIL

```

** Example 'RPTEX2S': REPEAT (with WHILE and UNTIL option)
*****
DEFINE DATA LOCAL
1 #X (I1) INIT <0>
1 #Y (I1) INIT <0>
END-DEFINE
*
REPEAT WHILE #X <= 5
    ADD 1 TO #X
    WRITE NOTITLE '=' #X
END-REPEAT
*
SKIP 3
REPEAT
    ADD 1 TO #Y
    WRITE '=' #Y
    UNTIL #Y = 6
END-REPEAT
*
END

```

Ausgabe des Programms RPTEX2S:

```

#X:    1
#X:    2
#X:    3
#X:    4
#X:    5
#X:    6

```

```

#Y:    1
#Y:    2
#Y:    3
#Y:    4
#Y:    5
#Y:    6

```

Äquivalentes Reporting-Mode-Beispiel: RPTEX2R.