

# DIVIDE

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiel

Verwandte Statements: ADD | COMPRESS | COMPUTE | EXAMINE | MOVE | MOVE ALL | MULTIPLY | RESET | SEPARATE | SUBTRACT

Gehört zur Funktionsgruppe: *Arithmetische Funktionen und Datenzuweisungen*

---

## Funktion

Mit dem Statement `DIVIDE` können Sie einen Operanden durch einen anderen dividieren.

### Division durch Null:

Wird eine Division durch Null (0) versucht, d.h. wenn der Divisor (*operand1*), also die Zahl durch die geteilt wird, 0 ist, wird entweder eine entsprechende Fehlermeldung oder als Ergebnis 0 ausgegeben, je nachdem wie der Session-Parameter `ZD` (der in der *Parameter-Referenz*-Dokumentation beschrieben ist) gesetzt ist.

## Syntax-Beschreibung

Für dieses Statement sind verschiedene Strukturen möglich:

- Syntax 1 — `DIVIDE` ohne `GIVING`-Klausel
- Syntax 2 — `DIVIDE`-Statement mit `GIVING`-Klausel
- Syntax 3 — `DIVIDE`-Statement mit `REMAINDER`-Option

Eine Erläuterung der in dem Syntax-Diagramm verwendeten Symbole entnehmen Sie dem Abschnitt *Syntax-Symbole*.

### Syntax 1 — `DIVIDE` ohne `GIVING`-Klausel

<code>DIVIDE [ROUNDED] operand1 INTO operand2</code>
--

Operanden-Definitionstabelle:





<i>operand2</i>	<p><b>Ergebnisfeld:</b></p> <p>Wird die GIVING-Klausel nicht benutzt, wird das Ergebnis in <i>operand2</i> ausgegeben. Das Ergebnisfeld kann ein Datenbankfeld oder eine Benutzervariable sein.</p> <p>Wenn <i>operand2</i> eine Konstante oder eine nicht änderbare Natural-Systemvariable ist, ist die GIVING-Klausel erforderlich.</p>
<b>ROUNDED</b>	<p>Wenn Sie das Schlüsselwort <b>ROUNDED</b> angeben, wird das Ergebnis gerundet.</p>
<b>GIVING</b> <i>operand3</i>	<p>Wenn das Schlüsselwort <b>GIVING</b> benutzt wird, wird <i>operand2</i> nicht geändert, und das Ergebnis wird in <i>operand3</i> ausgegeben.</p> <p>Wird ein Datenbankfeld als Ergebnisfeld verwendet, ändert sich durch die Division lediglich der programmintern verwendete Wert des Feldes; der in der Datenbank gespeicherte Wert des Feldes wird davon nicht berührt.</p> <p>Die Anzahl der Dezimalstellen für das Ergebnis der Division wird vom Ergebnisfeld (d.h. <i>operand2</i>, wenn keine GIVING-Klausel benutzt wird, oder <i>operand3</i>, wenn die GIVING-Klausel benutzt wird) ausgewertet.</p> <p>Informationen zur Genauigkeit der Ergebnisse siehe Abschnitt <i>Genauigkeit von Ergebnissen bei arithmetischen Operationen</i> im Leitfaden zur Programmierung.</p>
<b>REMAINDER</b> <i>operand4</i>	<p>Das Schlüsselwort <b>REMAINDER</b> bewirkt, dass der nach einer (ungerundeten) Division verbleibende Rest in <i>operand4</i> ausgegeben wird.</p> <p>Wenn <b>GIVING</b> und <b>REMAINDER</b> benutzt werden, kann keiner der vier Operanden ein Array-Bereich sein.</p> <p>Intern wird der Rest wie folgt berechnet:</p> <ol style="list-style-type: none"> <li>1. Der Quotient der Division von <i>operand1</i> und <i>operand2</i> wird berechnet.</li> <li>2. Der Quotient wird mit <i>operand1</i> multipliziert.</li> <li>3. Das Produkt dieser Multiplikation wird von <i>operand2</i> subtrahiert.</li> <li>4. Das Ergebnis dieser Subtraktion wird <i>operand4</i> zugewiesen.</li> </ol> <p>Für jeden dieser Schritte gelten die im Abschnitt <i>Genauigkeit von Ergebnissen bei arithmetischen Operationen</i> im Leitfaden zur Programmierung beschriebenen Regeln.</p>

## Beispiel

```

** Example 'DIVEX1': DIVIDE
*****
DEFINE DATA LOCAL
1 #A (N7) INIT <20>

```

```
1 #B (N7)
1 #C (N3.2)
1 #D (N1)
1 #E (N1) INIT <3>
1 #F (N1)
END-DEFINE
*
DIVIDE 5 INTO #A
WRITE NOTITLE 'DIVIDE 5 INTO #A' 20X '=' #A
*
RESET INITIAL #A
DIVIDE 5 INTO #A GIVING #B
WRITE 'DIVIDE 5 INTO #A GIVING #B' 10X '=' #B
*
DIVIDE 3 INTO 3.10 GIVING #C
WRITE 'DIVIDE 3 INTO 3.10 GIVING #C' 8X '=' #C
*
DIVIDE 3 INTO 3.1 GIVING #D
WRITE 'DIVIDE 3 INTO 3.1 GIVING #D' 9X '=' #D
*
DIVIDE 2 INTO #E REMAINDER #F
WRITE 'DIVIDE 2 INTO #E REMAINDER #F' 7X '=' #E '=' #F
*
END
```

Ausgabe des Programms DIVEX1:

```
DIVIDE 5 INTO #A           #A:      4
DIVIDE 5 INTO #A GIVING #B #B:      4
DIVIDE 3 INTO 3.10 GIVING #C #C:    1.03
DIVIDE 3 INTO 3.1 GIVING #D #D:     1
DIVIDE 2 INTO #E REMAINDER #F #E:  1 #F:  1
```