

DECIDE ON

```

DECIDE ON { FIRST } [VALUES] [OF] operand1
           { EVERY }
           {VALUES operand2 [[,operand2] ... [:operand2]] statement ...}
[ANY [VALUES] statement ... ]
[ALL [VALUES] statement ... ]
NONE [VALUES] statement ...
END-DECIDE

```

Dieses Kapitel behandelt folgende Themen:

- Funktion
- Syntax-Beschreibung
- Beispiele

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

Verwandte Statements: DECIDE FOR | IF | IF SELECTION | ON ERROR

Gehört zur Funktionsgruppe: *Logische Bedingungen*

Funktion

Das Statement `DECIDE ON` dient dazu, in Abhängigkeit vom Wert (bzw. von den Werten) einer Variablen eine oder mehrere Handlungen auszuführen.

Anmerkung:

Falls unter einer bestimmten Bedingung *keine* Handlung ausgeführt werden soll, geben Sie das Statement `IGNORE` in der betreffenden Klausel des `DECIDE ON`-Statements an.

Syntax-Beschreibung

Operanden-Definitionstabelle:

Operand	Mögliche Struktur	Mögliche Formate	Referenzierung erlaubt	Dynam. Definition
<i>operand1</i>	S A N	A U N P I F B D T L G O	ja	nein
<i>operand2</i>	C S A	A U N P I F B D T L G O	ja	nein

* Format B von *operand5*, *operand6*, *operand7* und *operand8* kann nur mit einer Länge von kleiner oder gleich 4 benutzt werden.

Syntax-Element-Beschreibung:

FIRST/EVERY	<p>Zu verarbeitender Wert:</p> <p>Mit einem dieser Schlüsselwörter geben Sie an, ob nur der erste gefundene Wert (FIRST) oder alle gefundenen Werte (EVERY) der Variablen verarbeitet werden sollen.</p>
<i>operand1</i>	<p>Kontrollfeld:</p> <p>Als <i>operand1</i> oder <i>operand2</i> geben Sie den Namen des Feldes an, dessen Werte geprüft werden sollen.</p>
<p>VALUES <i>operand2</i> [[,<i>operand2</i>] ... [:<i>operand2</i>]<i>statement</i> ...</p>	<p>Wert des Kontrollfeldes:</p> <p>Mit dieser Klausel geben Sie den Wert (<i>operand2</i>) des Kontrollfeldes an, sowie die <i>statements</i>, die ausgeführt werden sollen, wenn das Kontrollfeld diesen Wert hat.</p> <p>Sie können einen Wert, mehrere Werte oder einen Bereich von Werten angeben, vor denen als Option einer oder mehrere Wert/e stehen können.</p> <p>Werden mehrere Werte angegeben, müssen diese entweder mit dem Input-Delimiterzeichen (wie mit dem Session-Parameter ID definiert) oder mit einem Komma voneinander getrennt werden. Ein Komma darf hierzu allerdings nicht verwendet werden, falls das Komma als Dezimalkomma (mit dem Session-Parameter DC) definiert ist.</p> <p>Bei einem Bereich von Werten geben Sie, durch einen Doppelpunkt voneinander getrennt, den Anfangs- und den Endwert des Bereiches an.</p>

ANY statement	ANY-Klausel: Mit ANY geben Sie das (die) Statement(s) an, die ausgeführt werden sollen, wenn irgendeiner der in der VALUES-Klausel angegebenen Werte gefunden wird. Diese Statements werden zusätzlich zu den in der VALUES-Klausel angegebenen Statements ausgeführt.
ALL statement	ALL-Klausel: Mit ALL geben Sie das (die) Statement(s) an, die ausgeführt werden sollen, wenn alle in der VALUES-Klausel angegebenen Werte gefunden werden. Diese Statements werden zusätzlich zu den in der VALUES-Klausel angegebenen Statements ausgeführt. Die ALL-Klausel kann nur in Verbindung mit dem Schlüsselwort EVERY eingesetzt werden
NONE statement	NONE-Klausel: Mit NONE geben Sie das (die) Statement(s) an, die ausgeführt werden sollen, wenn keiner der angegebenen Werte gefunden wurde.
END-DECIDE	Das für Natural reservierte Wort END-DECIDE muss zum Beenden des DECIDE ON-Statements benutzt werden.

Beispiele

- Beispiel 1 — DECIDE ON-Statement mit FIRST-Option
- Beispiel 2 — DECIDE ON-Statement mit EVERY-Option

Beispiel 1 — DECIDE ON-Statement mit FIRST-Option

```

** Example 'DECEX3': DECIDE ON (with FIRST option)
*****
*
SET KEY ALL
INPUT 'Enter any PF key' /
      'and check result' /
*
DECIDE ON FIRST VALUE OF *PF-KEY

```

```

VALUE 'PF1'
  WRITE 'PF1 key entered.'
VALUE 'PF2'
  WRITE 'PF2 key entered.'
ANY VALUE
  WRITE 'PF1 or PF2 key entered.'
NONE VALUE
  WRITE 'Neither PF1 nor PF2 key entered.'
END-DECIDE
*
END

```

Ausgabe des Programms DECEX3:

Enter any PF key
and check result

Ausgabe nach Drücken von PF1:

Page 1

05-01-11 15:08:50

PF1 key entered.
PF1 or PF2 key entered.

Beispiel 2 — DECIDE ON-Statement mit EVERY-Option

```

** Example 'DECEX4': DECIDE ON (with EVERY option)
*****
DEFINE DATA LOCAL
1 #FIELD (N1)
END-DEFINE
*
INPUT 'Enter any value between 1 and 9:' #FIELD (SG=OFF)
*
DECIDE ON EVERY VALUE OF #FIELD
  VALUE 1 : 4
    WRITE 'Content of #FIELD is 1-4'
  VALUE 2 : 5
    WRITE 'Content of #FIELD is 2-5'
  ANY VALUE
    WRITE 'Content of #FIELD is 1-5'
  ALL VALUE
    WRITE 'Content of #FIELD is 2-4'
  NONE VALUE
    WRITE 'Content of #FIELD is not 1-5'
END-DECIDE
*
END

```

Ausgabe des Programms DECEX4:

ENTER ANY VALUE BETWEEN 1 AND 9: 4

Nach Eingabe und Bestätigung des Wertes 4:

Page 1

05-01-11 15:11:45

Content of #FIELD is 1-4
Content of #FIELD is 2-5
Content of #FIELD is 1-5
Content of #FIELD is 2-4