New and Hidden Statements

Introduction

The people at The Game Creators are always striving to improve DarkBASIC Pro and this includes adding new statements and modifying existing ones. Unfortunately, this means that any book on the language is incomplete almost from the moment it is published! To correct that problem, these pages give details of any new, hidden or updated statements that are not covered in the first edition of the Hands On DarkBASIC Pro books.

New Statements

The CEIL Statement

CEIL is short for ceiling. This strange statement returns the smallest possible integer which is not less than a given value.

For example, the expression

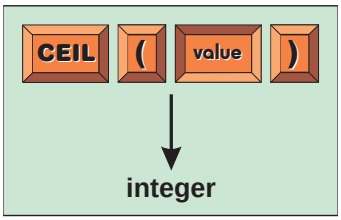
```
CEIL(12.9)
```

would return 13 since this is the smallest integer not less than 12.9.

The statement has the format shown in FIG-Sup05.

FIG-Sup05

The CEIL Statement



In the diagram:

value is a real number.

Hopefully, it is obvious that the expression

```
CEIL (9.1)
```

would return 10, but other examples may not be quite so obvious:

`CEIL(7.0)` returns 7

since 7 is not less than 7.0.

`CEIL(-2.3)` returns -2

since -3 (which you may have expected it to return) is larger than -2.3.

The FLOOR Statement

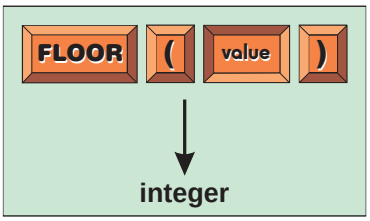
The FLOOR statement complements the CEIL statement since it reurns the largest integer not greater than a specified value. Hence

```
FLOOR( 12.9)    returns 12
FLOOR(9.1)      returns 9
FLOOR(7.0)      returns 7
FLOOR(-2.9)     returns -3
```

The format for the FLOOR statement is given in FIG-Sup06.

FIG-Sup06

The FLOOR Statement



In the diagram:

value is a real number.

The program in LISTING-Sup007 is a simple demonstration of the CEIL and FLOOR statements.

```
INPUT "Enter a number (end with -999): ", v#
WHILE v# <> -999
   PRINT "CEIL(",v#,") = ",CEIL(v#)
   PRINT "FLOOR(",v#,") = ",FLOOR(v#)
   INPUT "Enter a number ", v#
ENDWHILE
REM *** End program ***
END
```

Activity Sup017

Type in and test the program in LISTING-Sup007 (*cf.dbpro*).

Hidden Statements

The REVERSE$ Statement

The REVERSE$ statement can be used to reverse the order of the characters stored in a string variable. For example,

```
text$ = "ABCD"
REVERSE$ text$
```

would change the contents of text$ to "DCBA".

Notice that the string needs to be stored in a variable before calling REVERSE$. The expression

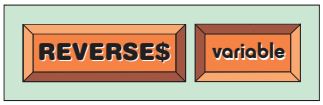
```
REVERSE$ "ABCD"
```

will compile but, since no variable is given, the result is not stored.

The format for the REVERSE$ statement is gven in FIG-Sup07.

FIG-Sup07

The REVERSE$ Statement



In the diagram:

variable	is a string variable. The contents of this variable will be reversed when the statement is executed.

Activity Sup018

Write a short program (*rev.dbpro*) that allows a string to be entered at the keyboard and then display the original string and the reversed version of the string.

Solutions

Activity Sup017

No solution required.

Activity Sup018

```
INPUT "Enter string : ", text$
PRINT "Text entered was : ", text$
REVERSE$ text$
PRINT "Text reversed is : ", text$
WAIT KEY
END
```