

DEF functionname(variablename,...,variablename)=expression

The functionname must start with FN. The variablenames in parentheses are optional; there may be no more than four. Suppose that in several places you need to determine the larger of two integer numbers. The following function will do this:

DEF FNLARGER%(X%,Y%)=X%\*(X%>Y%)+Y%\*(Y%>=X%)

If X% is greater than Y%, then X%>Y% yields one and Y%>=X% yields zero; thus the right side equals X%. If Y% is greater than or equal to X%, then the right side equals Y%. The values of X% and Y% are determined when the function is used. Here is an example:

```
10 G%=3%: A%=6%
20 PRINT FNLARGER%(G%*3%,A%)
30 DEF FNLARGER%(X%,Y%)=X%*(X%>Y%)+Y%*(Y%>=X%)
RUN
9
```

At line 20 we use the functionname. X% takes the value of G%\*3% (i.e., 9%), and Y% takes the value of A% (i.e., 6%). Since X% is greater than Y%, the function returns the value of 9%. Thus, at line 20, FNLARGER% is set to 9% and this is what is printed.

The variables X% and Y% are called dummy variables. Their values are set by the expressions in the parentheses following the use of the functionname in the program. These values hold, however, only in the definition of the function. If X% and Y% had been used elsewhere in the program, their value would have been unchanged by what happened to X% and Y% in the function. That is, the X% and Y% in the function would not be the same variables as X% and Y% used elsewhere in the program.

Functionnames can be real, integer (ends in %), or string (ends in \$). String functions always return string expressions; thus, by definition, the right side of a string function must be a string expression.

### Named Subroutines

Named subroutines are similar to ordinary subroutines, except they can have 0 to 4 dummy variables, and they are called in a different way. To call a named subroutine, just use its name; if dummy variables are used, set up expressions for each one. Subroutine names must end with a @ symbol. The command SUB followed by the subroutine name, defines the start of the routine. The command SUBEND defines the end of the named subroutine. If you want to exit from the middle of a named subroutine, you can use a RETURN command. Any number of RETURNS may be present; however, you may use only one SUBEND and it must be the last statement in the subroutine.