

VARIABLES AND DIMENSIONS

Variables and Dimensions are a way of storing information in STOS to use in your program. The two main pieces of information are words and numbers. Your program would decide if one of these equalled something and act upon it. Here's a simple example of a variable being used.

VARIABLES

```
10 print "Please type in a number"
```

```
20 input A
```

```
30 print "You choose the number";A
```

What happens here is that the computer labels a little box as 'A' and inserts the number you typed in at the input prompt into it. The number you typed is now stored in the computer's memory in a little box called 'A'. Line 30 looks in the box and prints its contents to the screen, which of course is the number you entered. 'A' is the name we have chosen for the variable and the input command is one way of putting information into it. Another way to put information into a variable is like this.

```
10 let NUMBER=100
```

```
20 print NUMBER
```

Line 10 tells the computer to get one of its boxes and stick a label on it and call it NUMBER, then puts the value '100' inside it. Note, the use of the command 'let' is optional and can be removed so your routine would be.

```
10 NUMBER=100
```

```
20 print NUMBER
```

Variables can be called anything you wish, using letters, words, and symbols, but you must make sure that there are no STOS commands in the name, or else you get something like this.....

```
10 sin GLE=10
```

If you used the word 'SINGLE' then STOS finds the command 'sin' and gets confused. Try different names until you find one that appears in capitals.

Not only numbers can be used in variables, so can words or letters. If you wanted to put a word into a variable then you must add a '\$' to the variable's name and the word must be put in between two quotes. For example.

```
10 NAME$="DEANO"
```

```
20 print "MY NAME IS ";NAME$
```

```
10 input "What is your name";NAME$
```

```
20 print "Hello There ";NAME$
```

You can also add variable values together which can be used in a game to add points to your score, for example.

```
10 SC=100
```

```
20 print "Your score is ";SC
```

```
30 print "Press a Key"
```

```
40 SC=SC+10 : goto 20
```

Line 40 tells the variable SC to equal the value of itself which is '100' and the add '10' to itself. The same applies to letters, usually known as strings of characters. For example.

```
10 input "What is your first name";NAME$
```

```
20 input "What is your second name";SURNAME$
```

```
30 A$=NAME$+" "+SURNAME$
```

```
40 print "Hello ";A$
```

DIMENSIONS

As you know, a variable is a little box inside the computer's memory. A dimension is a large box full of little boxes. Think of a cassette box, it's one big box with so many slots for your cassettes. A dimension allows you to store so many pieces of information inside one variable. Here's an example of how one could be used.

```
10 dim A(2) : rem Set up one variable box with two slots
```

```
20 A(1)=100 : A(2)=200
```

```
30 print A(1),A(2)
```

The same method can be used with strings.

```
10 dim NAME$(2)
```

```
20 NAME$(1)="ST" : NAME$(2)="PLUS"
```

```
30 print NAME$(1),NAME$(2)
```

Dimensions can be used the same way as variables; the only difference is that with dimensions you use the variable name plus the number of slots in the variable box....IE: dim SC(10).

Another type of dimension is the two-way dimension. This is setting up so many boxes with so many slots. Try this example.

```
10 dim NUMBER(5,10) : rem Set up five boxes each with ten slots.
```

```
20 NUMBER(1,1)=100 : rem Put `100` in box one, slot one
```

```
30 NUMBER(2,1)=500 : rem Put `500` in box two, slot one.
```

```
40 print NUMBER(1,1)
```

```
50 print NUMBER(2,1)
```

This is useful for directions in adventure games. For example, this routine sets up a dimension with five locations and four exits.

```
10 dim D$(5,4)
```

```
20 for X=1 to 5 : For Y=1 to 4
```

```
30 read D$(X,Y)
```

```
40 next Y : next X
```

```
50 data 2,3,4,5
```

```
60 data 1,2,3,4
```

```
70 data 5,2,4,3
```

```
80 data 7,6,2,1
```

```
90 data 2,4,1,2
```

I recommend you use clear variable names and plenty of REM statements in your program so you can keep track of what each variable and dimension does.