

DATA Step Information - ARRAY statement

The ARRAY statement defines a set of variables as elements of the array. Arrays can be either one- or two-dimensional. The ARRAY statement is used in the DATA step operations. The syntax of the ARRAY statement is:

ARRAY array-name {subscript} <\$> <array elements> <(initial values)> ;

Examples of one-dimensional arrays:

```
DATA a;
INPUT y x1 x2 x3 ;
ARRAY v{3} x1 x2 x3 ;
.
.
.
DATALINES;
```

```
DATA b;
INPUT response month1 $ month2 $ month3 $ month4 $;
ARRAY mo{4} month1 -- month4 ;
.
.
.
DATALINES;
```

In this second example, the double dash "-" can be used

For two-dimensional arrays:

```
ARRAY test{3,2} score1-score6;
```

produces an array with 3 rows and 2 columns, filling all the rows in order, beginning at the upper left

```
ARRAY test{1:3, 1:2} score1-score6;
```

is equivalent to the above ARRAY statement. Upper bound of 3 rows and 2 columns are specified.

```
ARRAY b{2} w x (7 4) ;
```

defines an array of two variables w and x. w has a starting or first value of 7, and x has a starting value of 4.

Using ARRAYS in a DO loop

Example:

```
DATA a ;
INPUT p q r s t u v x y z ;
ARRAY b{8} p--x ; *defines an array of 8 variables in the data set;
ARRAY c{8} d e f g h i j k ;
DO i=1 to 8 ;
    c{i} = b{i} * 2.54 ;    {or other SAS statements}
END;
DATALINES;
```

This could also be done by:

```
DATA a ;
INPUT p q r s t u v x y z ;

    *defines an array of 8 variables in the data set;
ARRAY b{8} p--x ;

    *defines an array of 8 new variables that are later given values;
ARRAY c{8} d e f g h i j k ;

DO OVER b;

    SAS statements

END;
DATALINES;
```