

## Two-dimensional arrays

### Defining a Multidimensional Array

To define a multidimensional array, you specify the number of elements in each dimension, separated by a comma.

In a two-dimensional array, the two dimensions can be thought of as a table of rows and columns.

`array array-name {row, column} array-elements;`

The first dimension in the ARRAY statement specifies the number of rows.

The second dimension specifies the number of columns.



So far, you have learned use one-dimensional arrays. In this video let's look at how to define and use two-dimensional arrays, which are a common type of multidimensional array.

Example: Two dimensional array

You can reference any element of the array by specifying the two dimensions.

```
array new{3,4} x1-x12;
```

```
new(2,3)=0;
```

x1	x2	x3	x4
x5	x6	x7	x8
x9	x10	x11	x12



Example:

Array new has row dimension as 3 and column dimension 4, which corresponds to 3 rows and 4 columns of the table, x1 – x12 are array elements

the row (2) and the column (3) corresponds to the array element x7 in the table,

`new(2,3)` reference x7

`new(2,3)=0;` performs an action (assign value 0) on the variable x7 by specifying the array reference new(2,3).

## SC statistical programming

When you define a two-dimensional array, the array elements are grouped in the order in which they are listed in the ARRAY statement. For example, the array elements x1 through x4 can be thought of as the first row of the table.

```
array new[3,4] x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12;
```

x1	x2	x3	x4
x5	x6	x7	x8
x9	x10	x11	x12

The elements x5 through x8 become the second row of the table, and so on.

```
array new[3,4] x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12;
```

x1	x2	x3	x4
x5	x6	x7	x8
x9	x10	x11	x12

