

Array Processing





You can use arrays to simplify programs that

- Perform repetitive calculations
- Create many variables with the same attributes
- Read data
- Rotate SAS data sets by making variables into observations or observations into variables
- Compare variables
- Perform a table lookup.





A SAS array

- Is a temporary grouping of SAS variables that are arranged in a particular order
- Is identified by an array name
- Exists only for the duration of the current DATA step
- Is not a variable

Each value in an array is

- Called an element
- Identified by a subscript that represents the position of the element in the array.

When you use an array reference, the corresponding value is substituted for the reference.





The ARRAY statement

- Defines the elements in an array.
- These elements will be processed as a group.
- You refer to elements of the array by the array name and subscript.

```
ARRAY array-name {subscript} <$> <length> <array-elements> < (initial-value-list) .;
```

- Must contain all numeric or all character elements
- Must be used to define an array before the array name can be referenced
- Creates variables if they do not already exist in the PDV
- Is a compile-time statement.

Creating One-Dimensional Arrays



Specifying the Array Name

To group the variables in the array, first give the array a name.

```
array sales{4} qtr1 qtr2 qtr3 qtr4;
```

Specifying the Dimension

- You must specify the dimension of the array.
- The dimension describes the number and arrangement of elements in the array

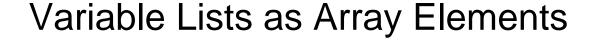
```
array sales{4} qtr1 qtr2 qtr3 qtr4;
```

The dimension of an array doesn't have to be the number of array elements

```
array sales (96:99) totals 96 totals 97 totals 98 totals 99;
```

Specifying Array Elements

When specifying the elements of an array, you can list each **variable name** that you want to include in the array





Specifying a Numbered Range of Variables

```
array sales{4} qtr1-qtr4;
```

When specifying a numbered range of variables,

- the variables must have the same name except for the last character or characters
- the last character of each variable must be numeric
- the variables must be numbered consecutively.
- Specifying All Numeric Variables

```
array sales(*) _numeric_;
```

Specifying All Character Variables

```
array sales(*) _character_;
```





General form, ARRAY reference:

array-name{index value}

where index value

- is enclosed in parentheses, braces, or brackets
- specifies an integer, a numeric variable, or a SAS numeric expression
- is within the lower and upper bounds of the dimension of the array.

Using the DIM Function in an Iterative DO Statement

 You can also use the **DIM** function to return the number of elements in the array.

DIM(array-name)

Creating Variables in an ARRAY Statement



General form, ARRAY statement to create new variables:

ARRAY array-name{dimension};

- Creating Default Variable Names
 - SAS creates default variable names by concatenating the array name and the numbers 1, 2, 3, and so on, up to the array dimension.

```
array WgtDiff{5};
```

If you prefer, you can specify individual variable names.

```
array WgtDiff{5} Oct12 Oct19 Oct26 Nov02 Nov09;
```

- Creating Temporary Array Elements
 - To create temporary array elements for DATA step processing without creating new variables, specify _TEMPORARY_ after the array name and dimension.

```
array goal{4} _temporary_ (9000 9300 9600 9900);
```

Temporary array elements do not appear in the resulting data set.