

## Arrays

An array is a group of variables, all of the same type that is referred to by a single name. The values in the group occupy consecutive locations in the computer's memory. An individual value within the array is called an array element; it is identified by the name of the array and a subscript pointing to a particular location within the array. For example, a(1) .

Arrays are very powerful tools for manipulating data in FORTRAN. We can manipulate and perform calculations with individual elements of arrays one by one, with whole arrays at once, or with various subsets of arrays.

### Declaring Arrays

An array may be of any type: real, integer, logical, or character. The type and size of an array are declared using a type declaration statement. For example, a real array voltage containing 16 elements could be declared as

```
REAL, DIMENSION (16) :: voltage
```

The DIMENSION attribute in the type declaration statement declares the size of the array being defined. The elements in array voltage would be addressed as voltage(1), voltage(2), and so forth, up to voltage(16).

Arrays may be declared with more than one subscript; so they may be organized into two or more dimensions. The number of subscripts declared for a given array is called the rank of the array. Array voltage is rank-1 array. The number of elements in a given dimension of an array is called the extent of the array in that dimension. The extent of the first (and only) subscript of array voltage is 16. The shape of the array is the combination of its rank and the extent of the array in each dimension. The size of the array is the total number of elements declared in that array. The size of the array voltage is 16.

An array constant is an array consisting entirely of constants. It is defined by placing the constant values between special delimiters called array constructors. For example, an array constant containing five integer elements is

```
(/1, 2, 3, 4, 5 /)
```

*Always initialize the elements in an array before you use them.*

You can use one of three techniques to initialize the elements in an array

1. Initial values may be assigned to the array using assignment statements.
2. Initial values may be declared in type declaration statements and loaded into the array at the compilation time.
3. Initial values may be read into the array using FORTRAN 90/95 READ statements.

- Example 1,           REAL, DIMENSION(10) :: array1  
                   DO i = 1,10  
                   Array1(i) = 0.0  
                   END DO
- Example 2,           REAL, DIMENSION(10) :: array1  
                   array1 = (/0.,0.,0.,0.,0.,0.,0.,0.,0.,0./)
- Example 3,           INTEGER, DIMENSION(5) :: array2=(/ 1, 2, 3, 4, 5 /)
- Example 4,           INTEGER, DIMENSION(5) :: array2 = ( / (i, i =1,5) /)
- Example 5,           INTEGER, DIMENSION(25) :: array4 = ( / ((0, i=1,4), 5\*j , j=1,5) /)  
                   The pattern of values produced by these nested loops is  
                   0, 0, 0, 0, 5, 0, 0, 0, 0, 10, 0, 0, 0, 0, 15, .....

*Always use named constants to declare the sizes of arrays in a FORTRAN program to make the array sizes easy to change.*

Examples of Array Elements:

```
WRITE (*,100) a(1), a(2), a(3), a(4), a(5)
100 FORMAT (1X, 'a =', 5F10.2)
```

```
WRITE(*,100) (a(i), i = 1,5)
100 FORMAT (1X, 'a =', 5F10.2)
```

Examples of Declaring Rank-2 Arrays:

1. REAL, DIMENSION(3,6) :: sum  
 The legal values of the first subscript are 1 to 3, and the legal values of the second subscript are 1 to 6. Any other subscript values are out of bounds.
2. INTEGER, DIMENSION(0:100,0:20) :: hist  
 It consists of 101 rows and 21 columns, for a total of 2121 elements. The legal values of the first subscript are 0 to 100, and the legal values of the second subscript are 0 to 20. Any other subscript values are out of bounds.

*Do not use arrays to solve a problem unless they are actually needed.*