

## Syntax and Operations

---

Most languages refer to an element of an array by appending a subscript—usually delimited by square brackets—to the name of the array: `Array[3]`

In some languages one declares an array by appending subscript notation to the syntax that would be used to declare a scalar. In C:

```
char upper[N];
```

In C, the lower bound of an index range is always zero: the indices of an  $n$ -element array are  $0 \dots n - 1$ .

In some languages, one can also declare a **multi-dimensional** array (array of arrays) by using the `array` constructor more than once in the same declaration.

In C, one must also declare an array of arrays, and use two-subscript notation, but C's integration of pointers and arrays means that slices are not supported.

```
char matrix[N][N];
```

given this definition, `matrix[3][4]` denotes an individual element of the array, but `matrix[3]` denotes a **reference**, to either the third row of the array or to the first element of that row, depending on context.

## Slices and Array Operations

---

A **slice** or **section** is a rectangular portion of an array. Many scripting languages provide extensive facilities for slicing.

In most languages, the only operations permitted on an array are selection of an element, and assignment. A few languages allow arrays to be compared for equality.