Two-Dimensional Arrays

One-Dimensional Arrays

Recall: An array is a collection of one type of data (e.g., integer, string) that is used for a single purpose (e.g., grades, addresses).

Each box is called an element of the array, and the position of each element is the index.

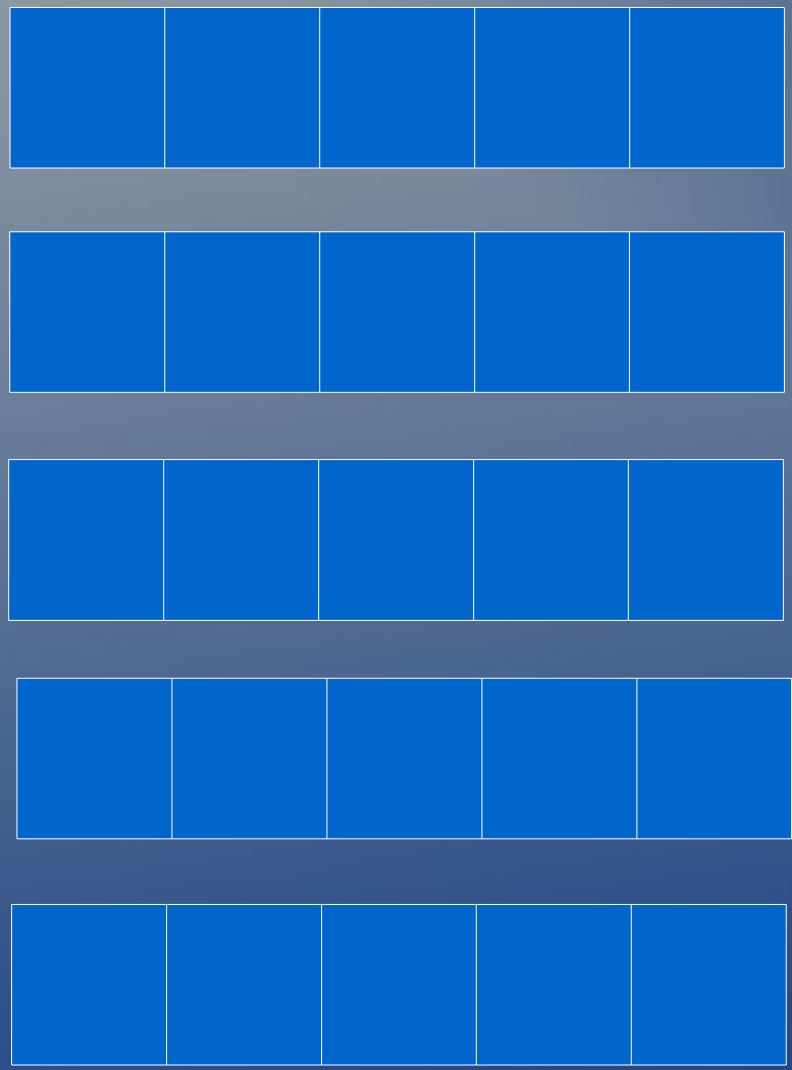
an array with 5 elements

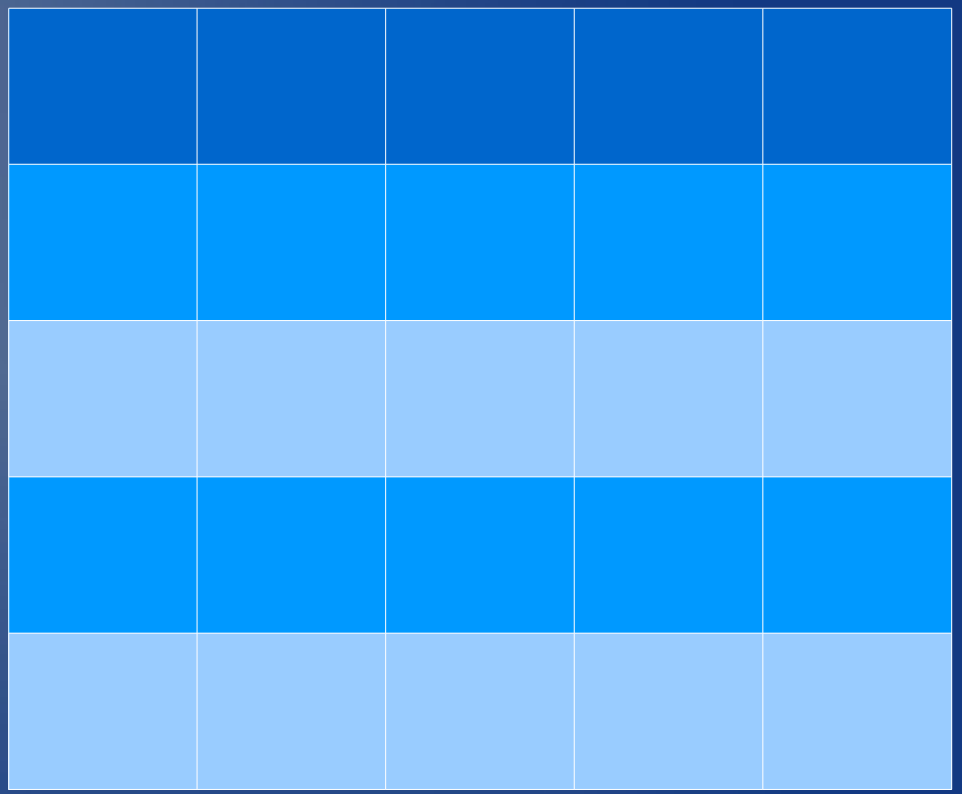
Two-Dimensional Arrays

A two-dimensional array is like having many one-dimensional arrays stacked on top of each other. Instead of a line of boxes, it forms a grid.

Otherwise, all of the regular rules for arrays apply:
1. All elements are of the same data type.
2. All data has the same theme or purpose.
3. The array should be initialized before it is used.

A 2-D array is several 1-D arrays joined together



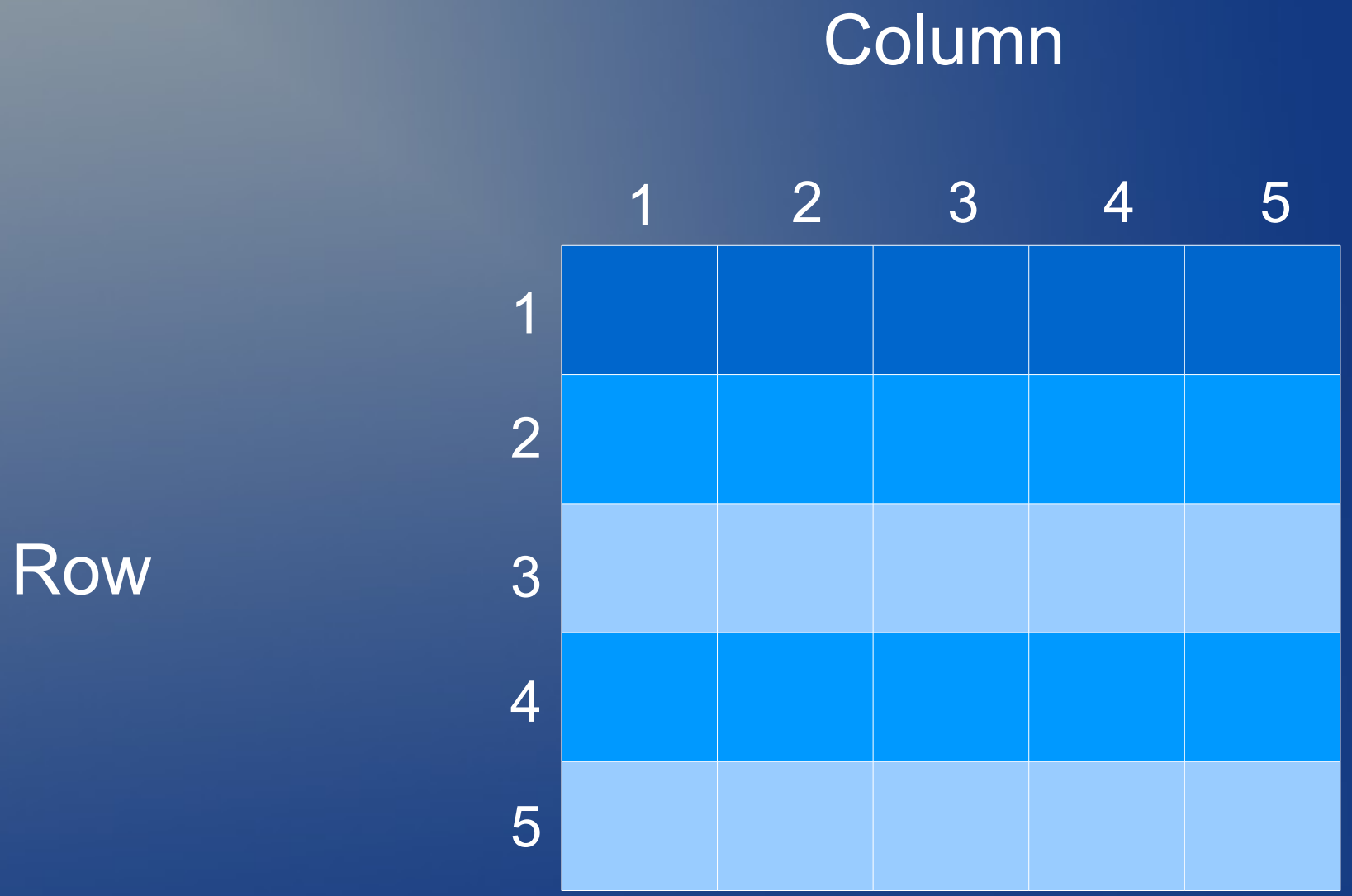


Referencing Cells in an Array

The elements of an array are contained in cells. In a one-dimensional array, each cell has a single index.

To reference (or access) the cells of a 2-D array, each cell must have two indexes – a row and a column.

Rows & Columns in a 2-D Array



Declaring 2-D Arrays in Turing

var name : array low1 .. high1, low2 .. high2 of dataType

name – the name of the array

low1 – the lower index value of the rows
high1 – the upper index value of the rows

low2 – the lower index value of the columns
high2 – the upper index value of the columns

dataType – integer, string, real, etc...

Traversing a 2-D Array

for loops are still the best method for traversing the array. With a second dimension, we will need a second loop. In general, we need a loop within a loop, or a nested loop

```
var arr : array 1..5, 1..10 of int
%                    rows  columns

for i : 1 .. 5              % number of rows
   for j : 1 .. 10          % number of columns
      arr(i,j) := 0
   end for
end for
```