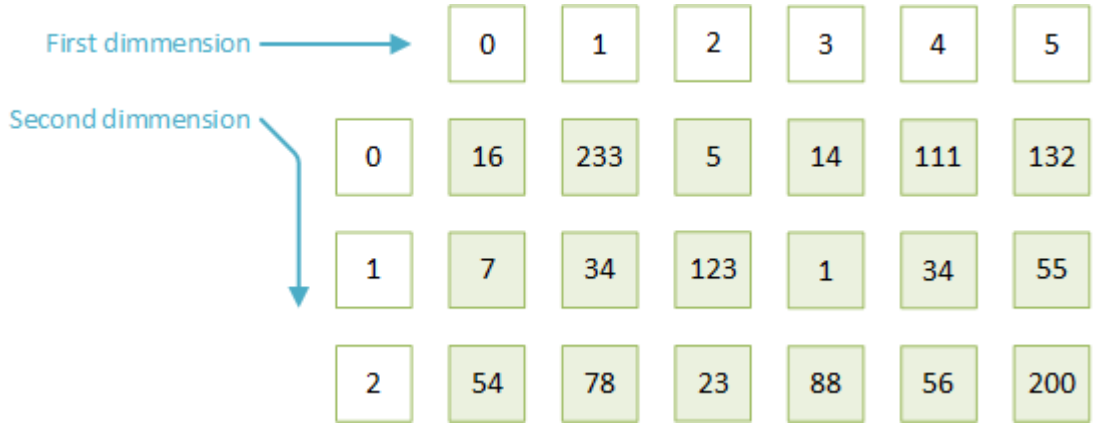
Array

An **array** is used to group data of the same type into blocks that can be easily addressed, i.e., indexed.

Arrays can be 1-, 2-, or 3-dimensional, or even 6-dimensional. The following example illustrates the structure of 2- and 3-dimensional arrays:



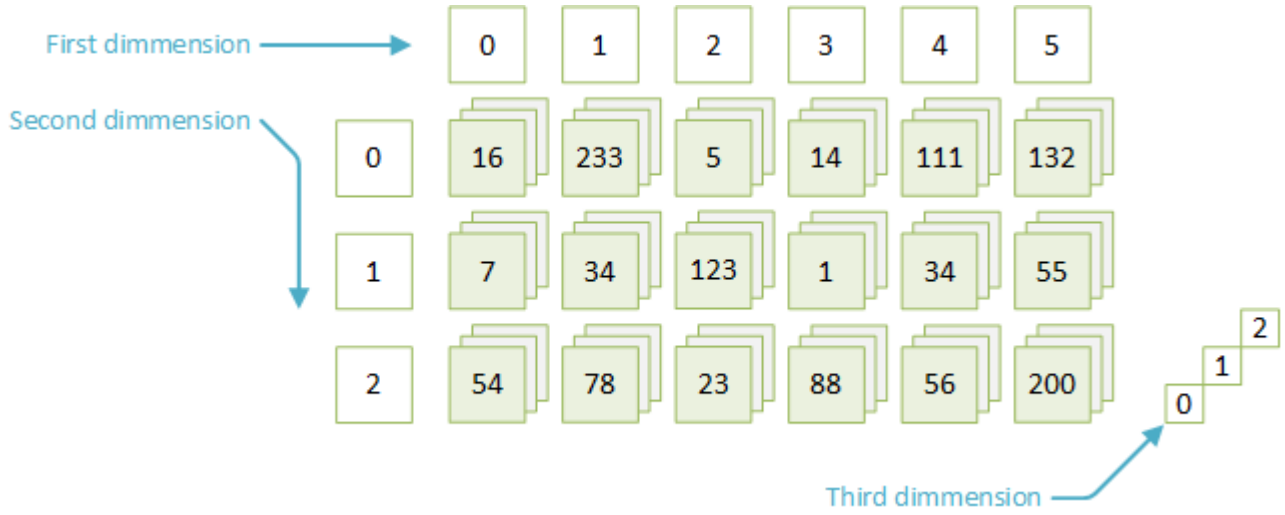
The image above displays a two-dimensional array of type “byte.” The first index represents the rows, while the second represents the columns. The value range of a byte is 0 to 255, so only values within this range are allowed. In the example above, the program's type definition is as follows:

```
arry : Array[0..5, 0..2] of Byte;
```

The assignment is displayed in the code like this:

```
tomb[3, 1] := 1;
```

The indexing of a three-dimensional array can be illustrated as follows:



In this case, the above assignment can be defined in the program as follows:

```
tomb[3, 1, 0] := 1;
```

Array of struct

The elements of the array are always homogeneous, meaning their types cannot vary. However, there can be multiple instances of a single type within a single array if we define a Struct type as an array element. The hydraulic motors described as an example in Struct can also be defined as an array:

motors

Name	Data type	Start value	Monitor value	Comment
tomb	Array[0..4] of Struct			
tomb[0]	Struct			
motorId	String	'1 4LAC01'		1 4LAC01 Hydraulicpumpe
voltage	Real	402.0		1 4LAC01 CE001 (V)
current	Real	5.2		1 4LAC01 CE002 (A)
mode	Byte	3		1 4LAC01 CE003 (On, Off, Auto)
status	Byte	1		1 4LAC01 CE004 (Run, Stop, Error)
local	Byte	1		1 4LAC01 CE005 (remote, local)
tomb[1]	Struct			
motorId	String	''		1 4LAC01 Hydraulicpumpe
voltage	Real	0.0		1 4LAC01 CE001 (V)
current	Real	0.0		1 4LAC01 CE002 (A)
mode	Byte	16#0		1 4LAC01 CE003 (On, Off, Auto)
status	Byte	16#0		1 4LAC01 CE004 (Run, Stop, Error)
local	Byte	16#0		1 4LAC01 CE005 (remote, local)
tomb[2]	Struct			
tomb[3]	Struct			
tomb[4]	Struct			

In this case, I specified the type of the four-element array as “Struct”. Here, a field opens under the name of the first array element (tomb[0]), where the Struct's elements can be defined. It is important that the array is homogeneous, meaning the structure can only be set for the first element; the other elements will be copies of it without the ability to modify the structure (values, of course, can change). In the example above, the value assignment will look like this (the DB name is *“motors”*):

```
"motors".tomb[1].current := 32.2;
```

More information:
TIA Datatypes: S7 data types summary table