**Multidimensional Arrays**

As said before, arrays can contain lots of different data types. So far, we’ve used primitive types like strings and numbers:

**const myArray = [ 10, 20, 30, 40 ];**

But arrays can also contain other arrays. An array that contains arrays as its elements is called a 2-Dimensional Array. These are the most common you’ll encounter. If there are more layers of nested elements of arrays, they would be called 3-Dimensional (an array of arrays of arrays) or 4-Dimensional etc. An array that is an element inside of another array can also be referred to as a nested array.

**const multiArray = [ //** *Encompassing array*

**[ 1, 2, 3 ], //** *1st element array*

**[ 4, 5, 6 ], //** *2nd element array*

**[ 7, 8, 9 ] //** *3rd element array*

**];**

Accessing elements in these multidimensional arrays is a bit more complicated but the rules you learned about simple arrays still apply. If you want to get the **[1, 2, 3]** array in **multiArray** you just need to access **multiArray’s** first element as you have done before:

**multiArray[ 0 ]; //** *this is* **[ 1, 2, 3 ]** *as [ 0 ] represents the element in the first index position*

**multiArray[ 2 ]; //** *this is* **[ 7, 8, 9 ]** *as [ 2 ] represents the element in the third index position*

However, if you want to access an element inside the nested arrays you need to add another pair of square brackets and use the correct index.

**multiArray[ 0 ][ 0 ]; //** *this is* **1** *as the first* [0] *selects the 1st array and the second* [0] *selects the 1st item*

**multiArray[ 0 ][ 1 ]; //** *this is* **2** *as the first* [0] *selects the 1st array and the second* [1] *selects the 2nd item*

**multiArray[ 1 ][ 2 ]; //** *this is* **6** *as the first* [1] *selects the 2nd array and the second* [2] *selects the 3rd item*

**multiArray[ 2 ][ 0 ]; //** *this is* **7** *as the first* [2] *selects the 1st array and the second* [0] *selects the 2nd item*

This is what multiArray would look like if the elements were replaced with their index positions.

**const multiArray = [ //** *Encompassing array*

**[ [0, 0], [0, 1], [0, 2] ],**

**[ [1, 0], [1, 1], [1, 2] ],**

**[ [2, 0], [2, 1], [2, 2] ]**

**];**

**Loops and Multidimensional Arrays**

In order to loop through all the elements contained inside a multidimensional array, you will need to use nested loops – loops inside of loops.

**const multiArray = [ //** *Encompassing array*

**[ 1, 2, 3 ], //** *multiArray[0]*

**[ 4, 5, 6 ], //** *multiArray[1]*

**[ 7, 8, 9 ] //** *multiArray[2]*

**];**

**for ( let arr = 0; arr < multiArr.length; arr += 1 ) {**

**//** *This area loops through each array*

**//** *multiArr[ arr ]*

**for ( let item = 0; item < multiArr[ arr ].length; item += 1 ) {**

**//** *This area loops through each element*

**//** *multiArr[ arr ][ item ]*

**}**

**}**