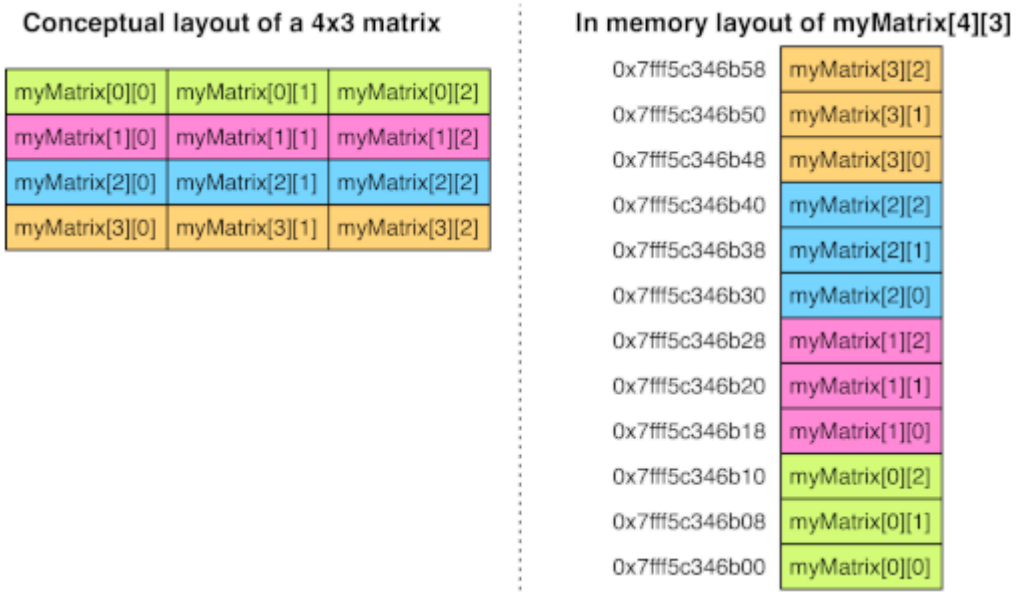
Indexing | Coursera

coursera.org/learn/pointers-arrays-recursion/supplement/hFuoQ/indexing

Indexing

Indexing

If we were to write *myMatrix*[*i*] (where i is some integer type), then we would expect that expression to evaluate to the ith element of *myMatrix* according to the rules that we learned perviously. For example, if we wrote *myMatrix*[*2*], we would expect that to evaluate to the 2nd element of *myMatrix*, which is the three blue boxes in the figure below. This element is an array of three doubles, so we would expect the type to be **double** *, and as the first element of that array is at *0x7fff5c346b30* (in this particular example), we would expect that to be the value of the expression (as we represent arrays by a pointer to their first element). If you expect all of these things (based on what you have already learned), you would be correct.



We may wish to index the two-dimensional array twice, such as *myArray*[*2*][*1*]. When the program evaluates this expression, it will first evaluate *myArray*[*2*], obtaining a pointer to the 3-element array which is the 2nd element of *myArray*. Then, it will index that array (which is an array of **double**s), and evaluate to a **double**. Of course, *myArray*[*2*][*1*] is an lvalue, as it names a particular box, so we can use it on the left side of an assignment, *e.g, myArray*[*2*][*1*] = *3.14*;. However, we should note that *myArray*[*2*] is not an lvalue, just like *myArray* is not an lvalue. The pointer that *myArray*[*2*] evaluates to is not actually stored anywhere, it is just calculated by pointer arithmetic from *myArray*.