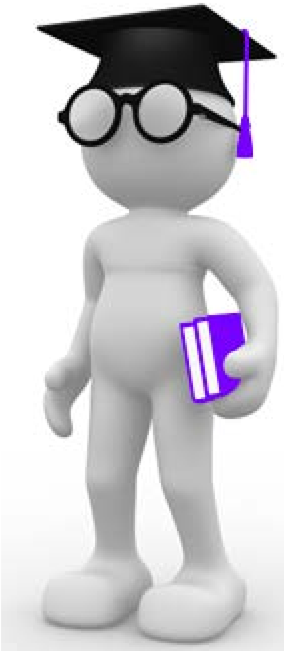
Multidimensional Arrays and Bounds

Simon Robinson
http://TechieSimon.com
@TechieSimon

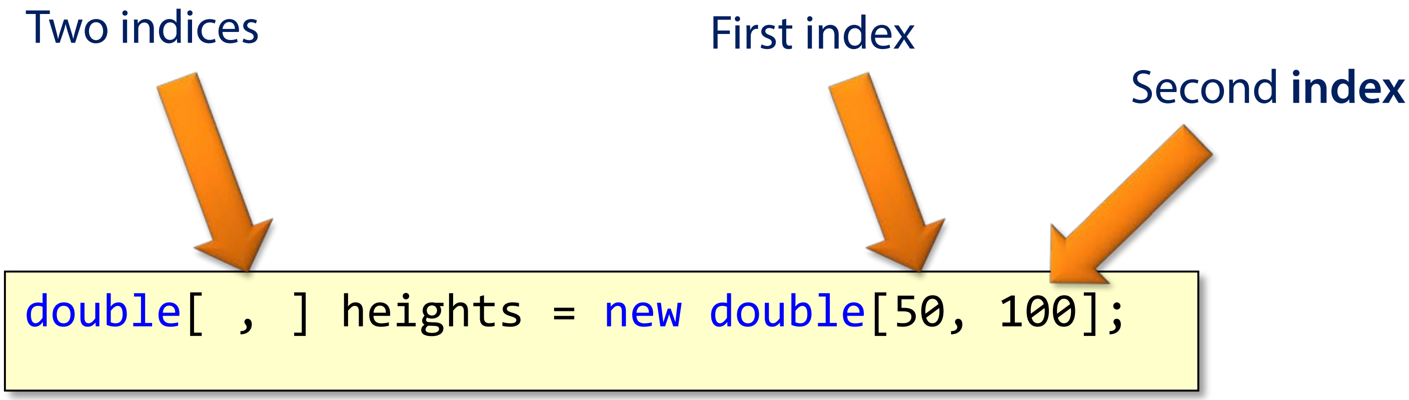
pluralsight
hardcore developer training

Module Overview

- **Multidimensional Arrays**
  - □ What they are
- **Rank and Bounds**
  - □ Different array index ranges
- **Jagged Arrays**
  - □ What they are



Multidimensional Arrays



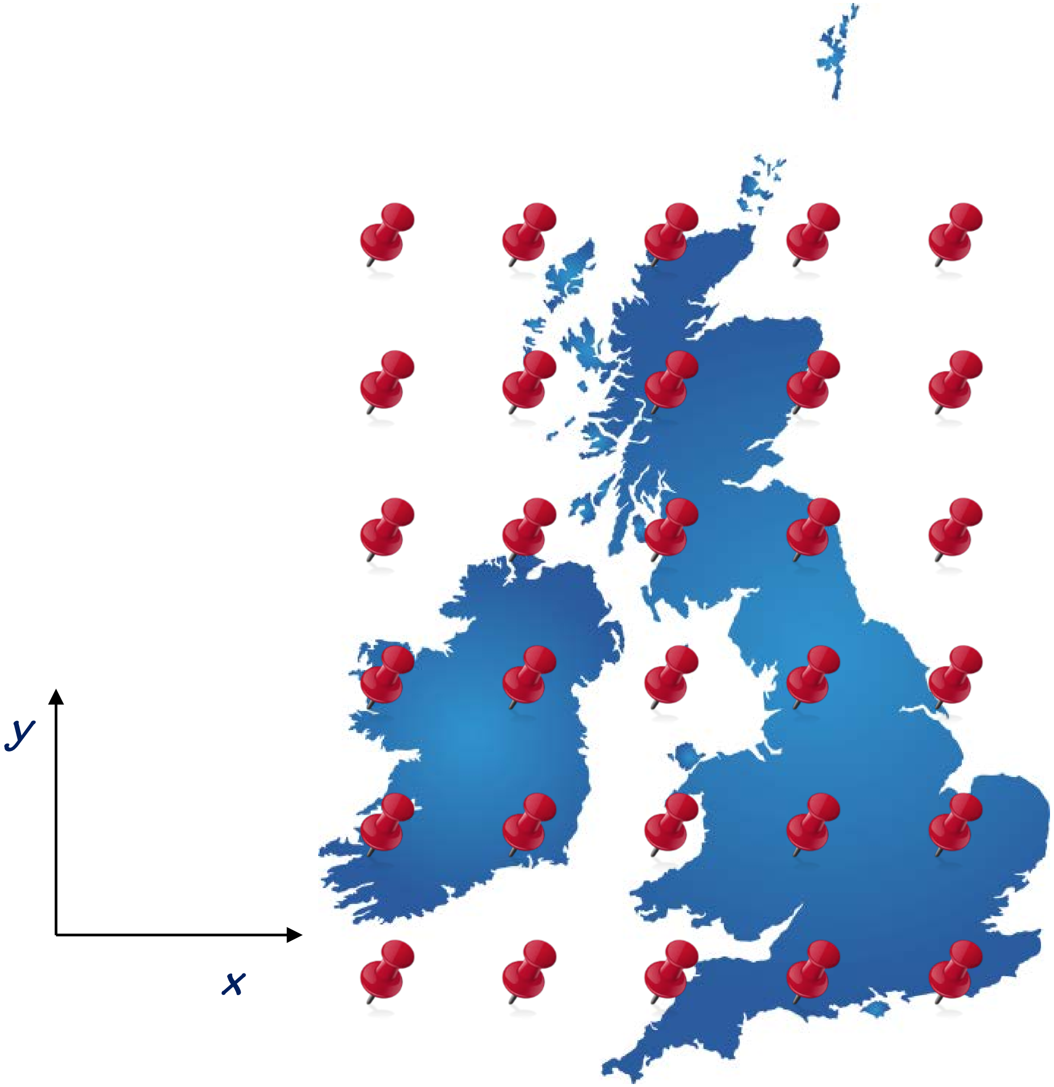
Two indices

First index

Second **index**

```
double[ , ] heights = new double[50, 100];
```

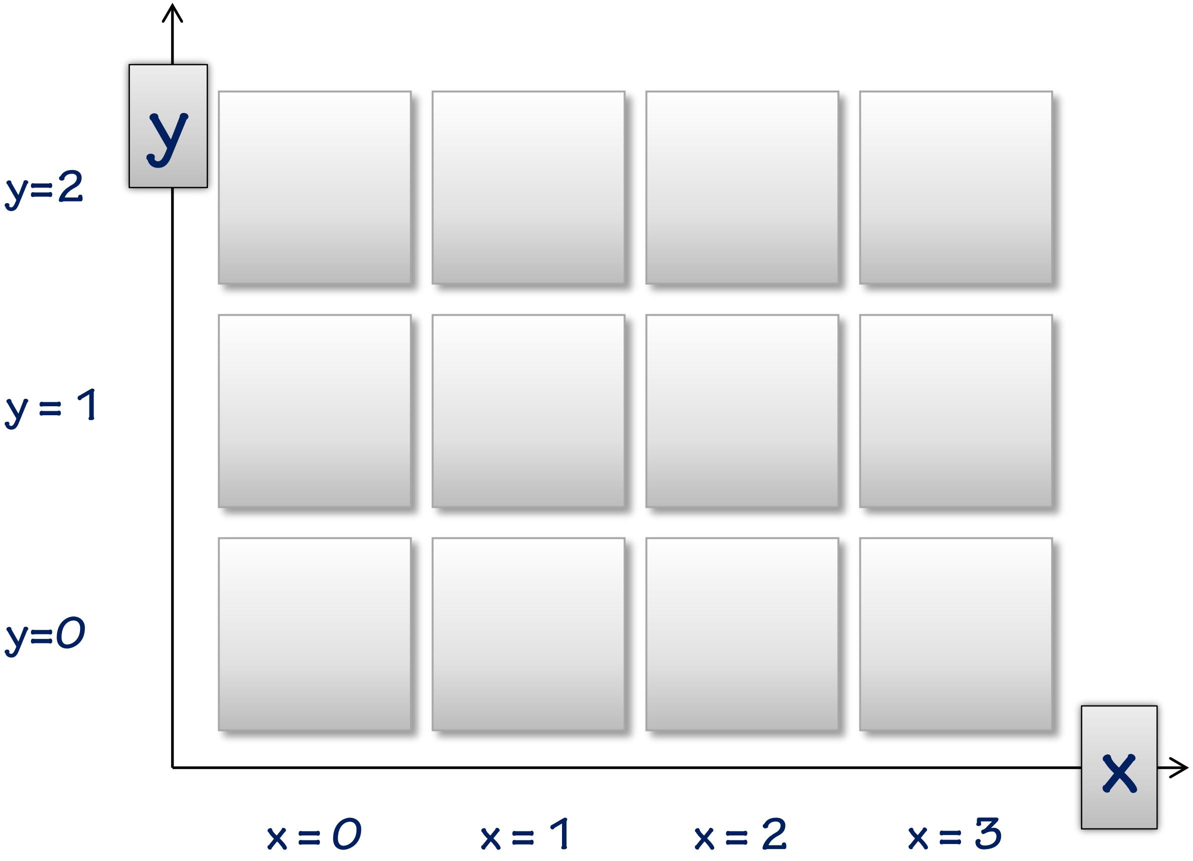
Only for arrays!



y
x

Collection of `float`

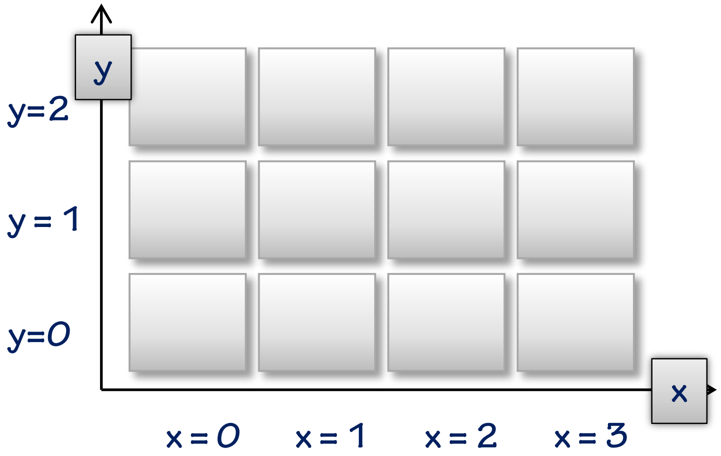
With two integers to look-up elements



Multiple Indices

```
float[,] floatArray2d;
```

Comma
tells the compiler that
elements should be accessed
with two indices



```
float[,,] floatArray3d;
```

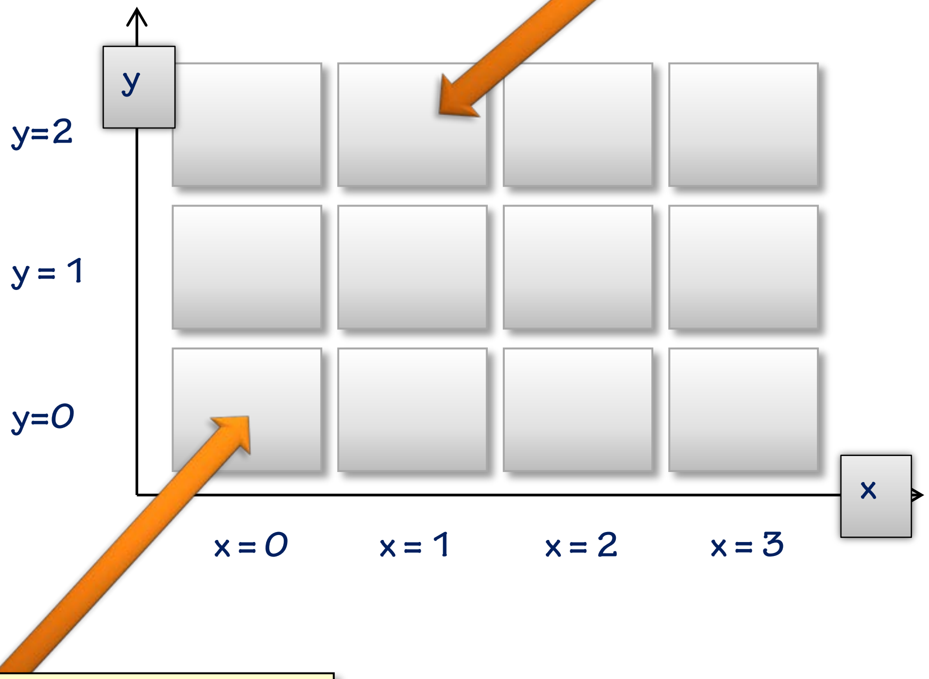
Two commas indicate
three indices etc.

No. of indices is
called:
the **dimension**
or the **rank**

Code Demo

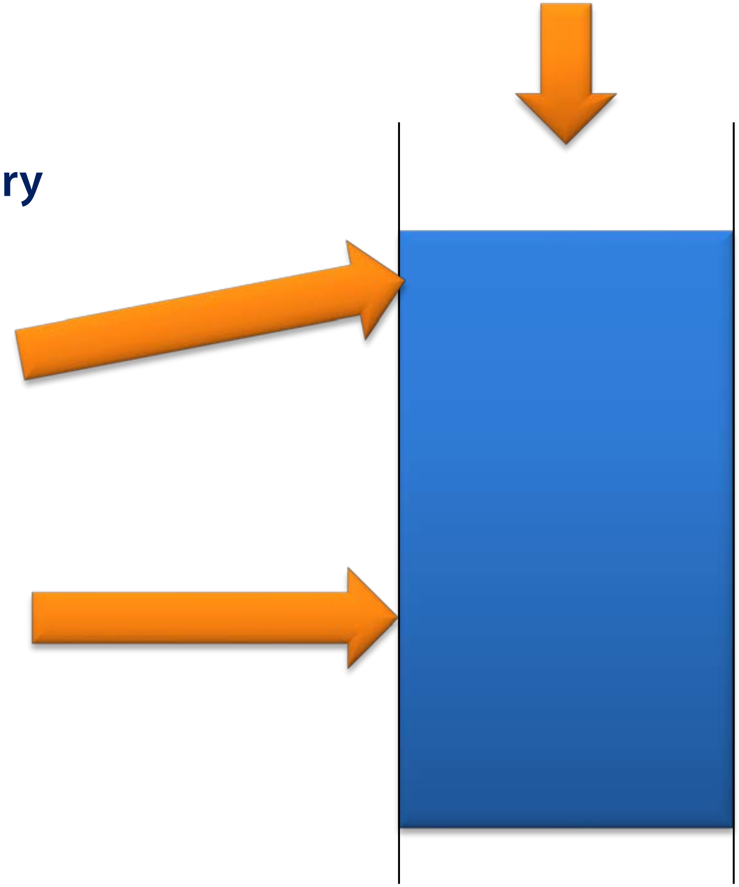
Need two integers to look up an element

```
tempsGrid[1,2];
```



```
tempsGrid[0,0];
```

```
float[,] tempsGrid = new float[4, 3];
```



Single memory block

Elements still go
one after the other
in the block of memory

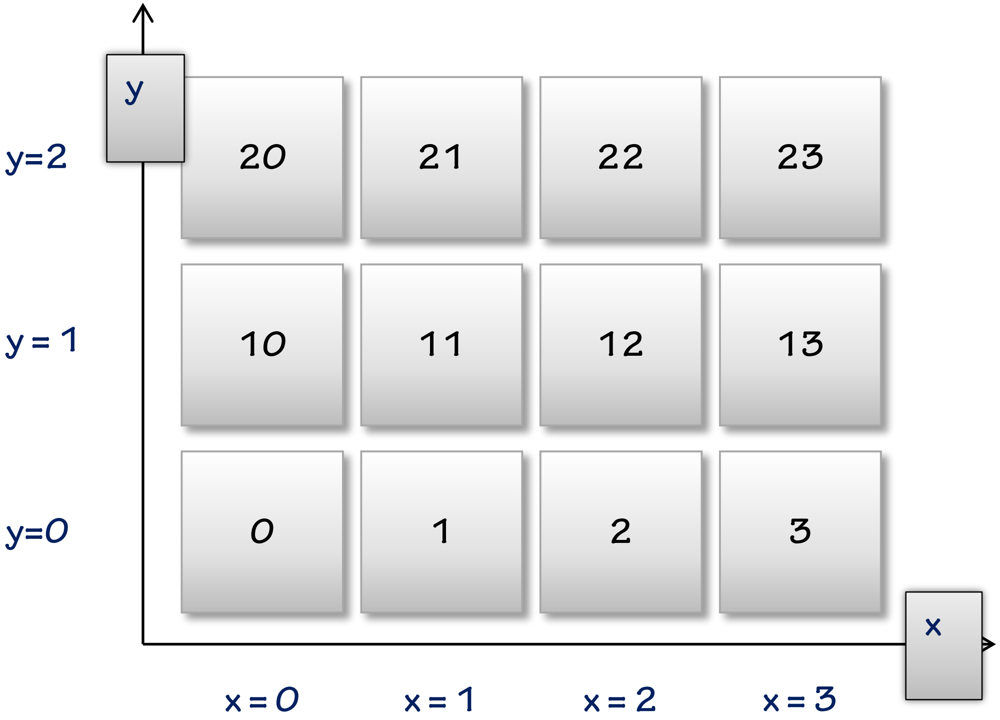
`tempsGrid[0,0]`

`tempsGrid[1,2]`

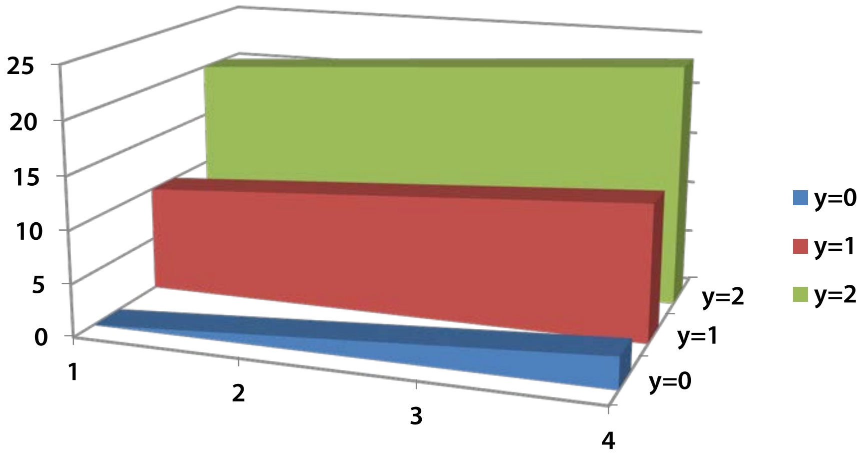
etc.

We will initialize the array with these values

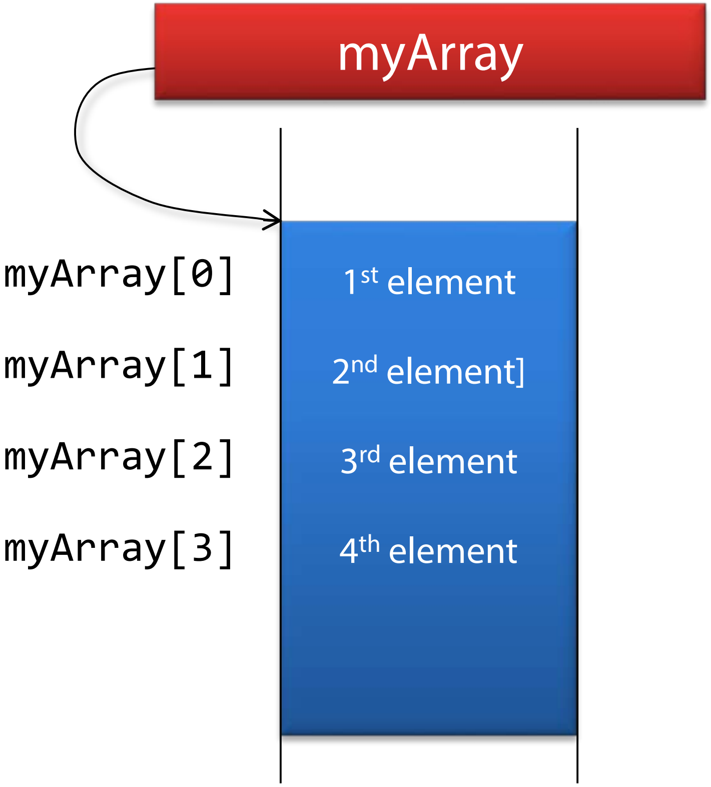
temperature = x + 10*y



	x = 0	x = 1	x = 2	x = 3
y=2	20	21	22	23
y = 1	10	11	12	13
y=0	0	1	2	3



Zero-based index



myArray

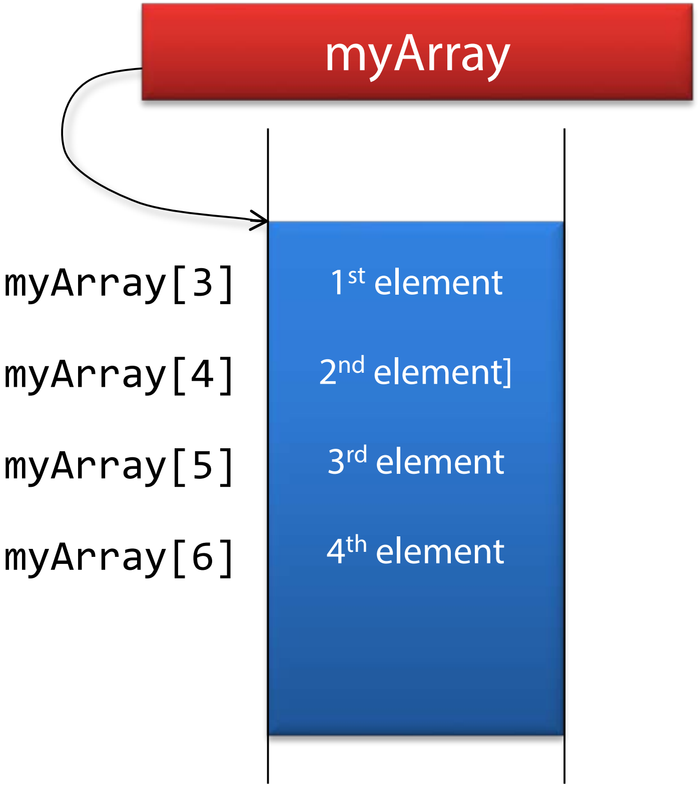
myArray[0] 1st element

myArray[1] 2nd element]

myArray[2] 3rd element

myArray[3] 4th element

Index starts at 3



myArray

myArray[3] 1st element

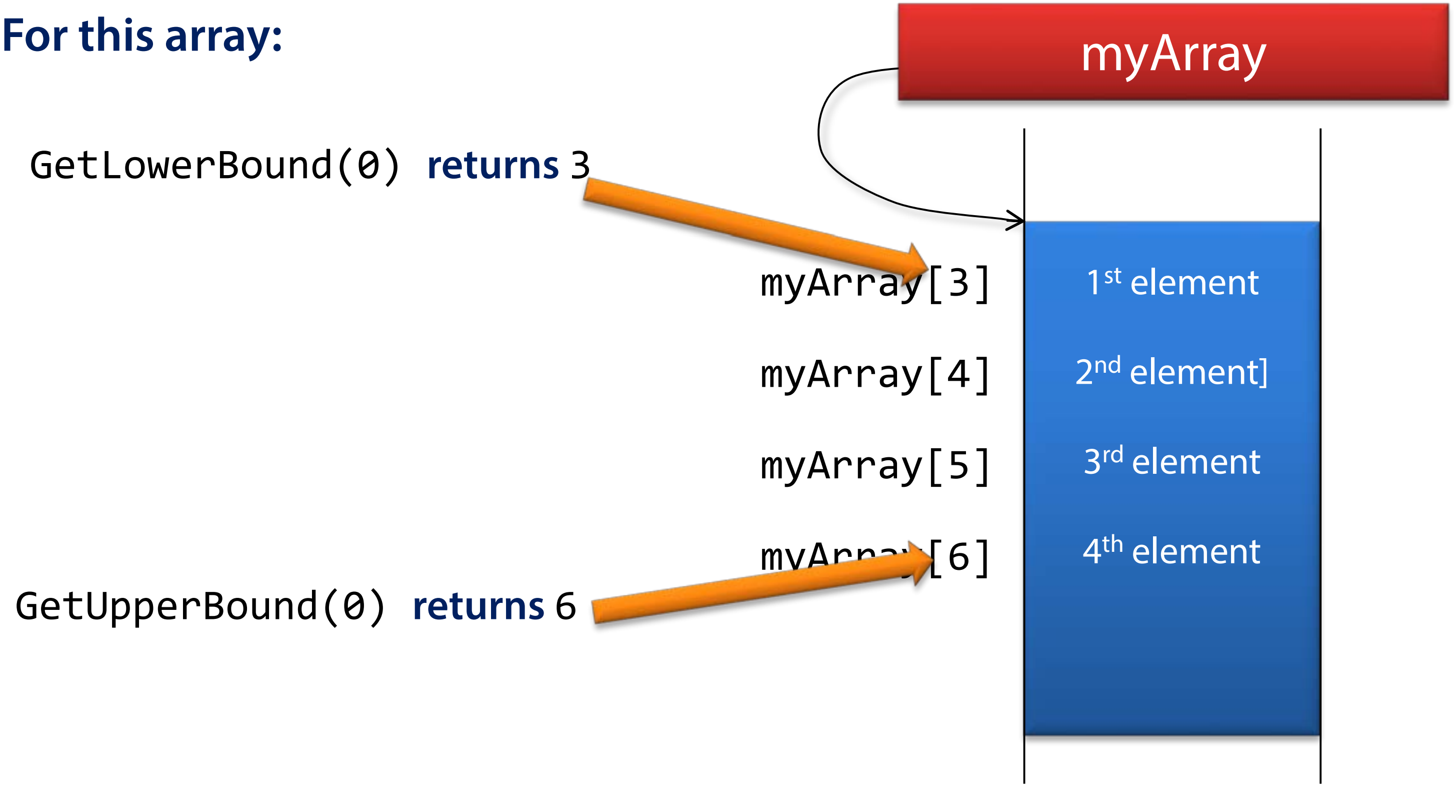
myArray[4] 2nd element]

myArray[5] 3rd element

myArray[6] 4th element

Index starts at 3

For this array:



GetLowerBound(0) **returns** 3

GetUpperBound(0) **returns** 6

Code Demo

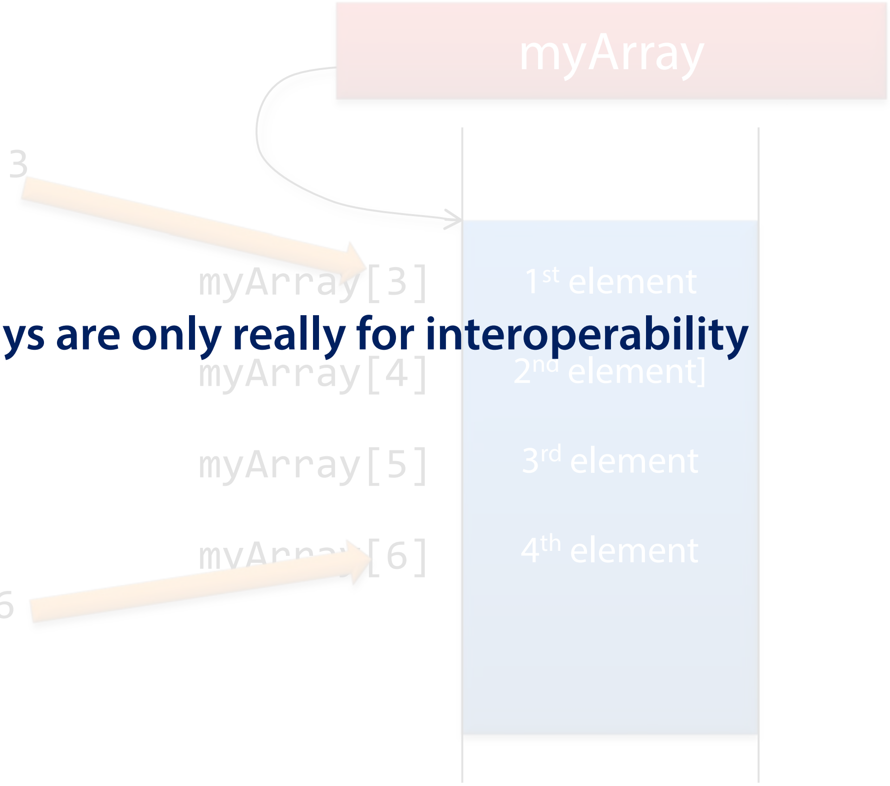
Index starts at 3

Why haven't I mentioned this before?

For this array:

Because this is very rare

`GetLowerBound(0)` **returns** 3



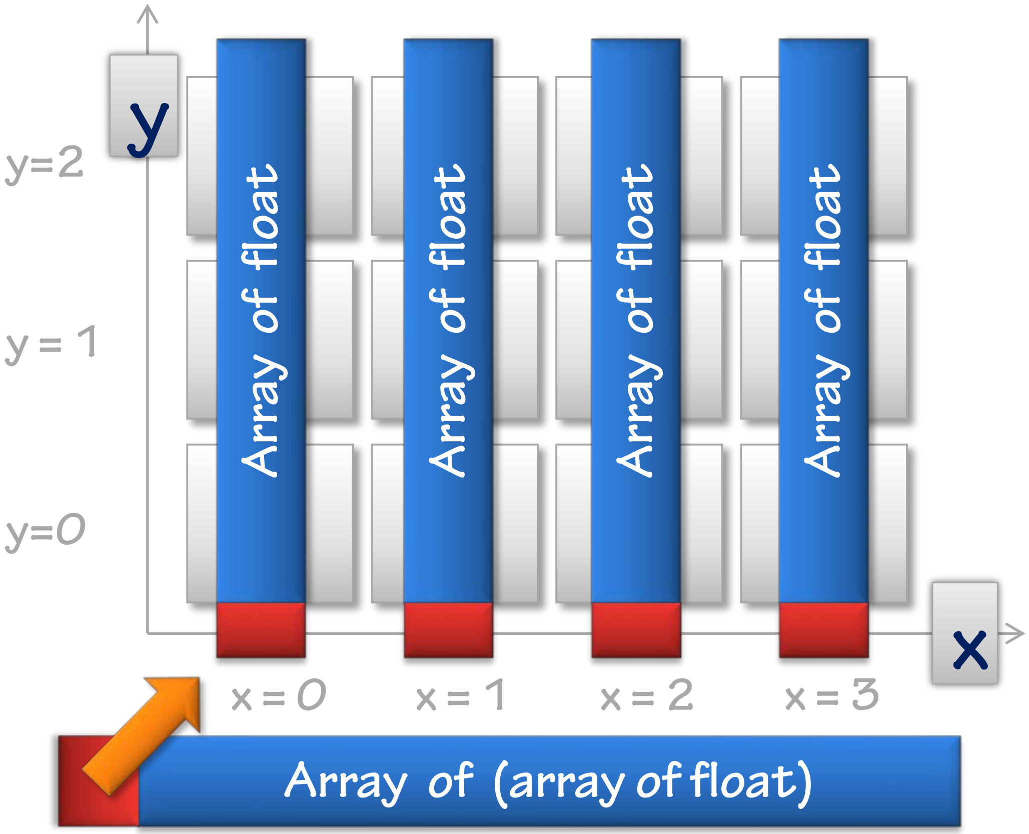
Non-zero-based arrays are only really for interoperability

`UpperBound(0)` **returns** 6

Jagged Arrays

Concept: An array of arrays

Each element of the outer array
is another array



```
float[,] floatArray2d;
```

Comma means this is a
2-dimensional array of floats

```
float[][] floatArrayJagged;
```

Additional square brackets
mean this is an
array of arrays of floats

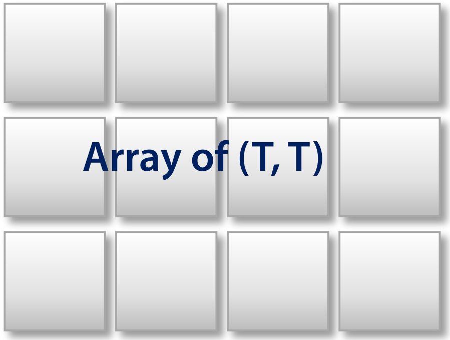
More than 2 indices…

```
float[,,] floatArray3d;
```

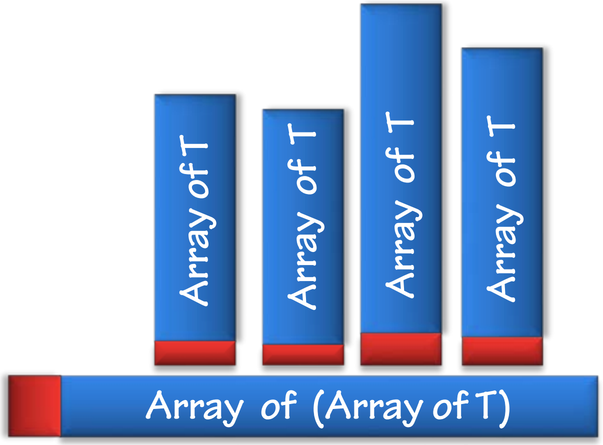
```
float[][][] floatArrayJagged3;
```

Code Demo

Jagged vs Multidimensional

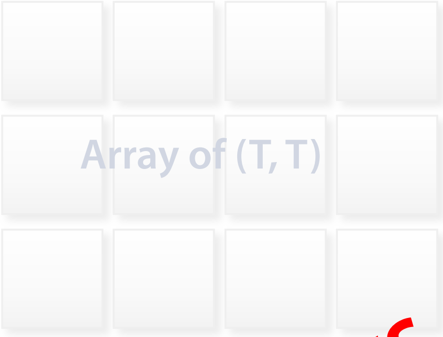


Simpler and more lightweight
– only one array



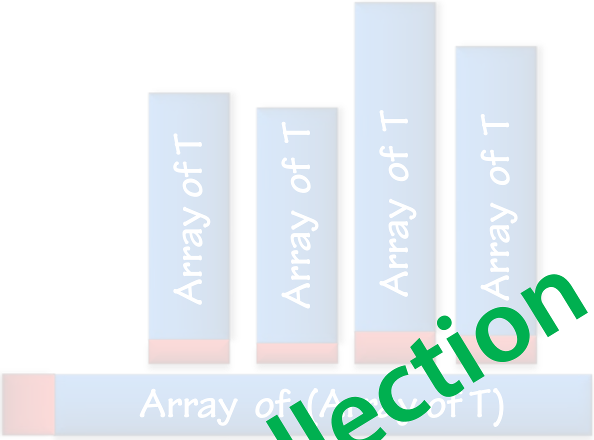
More flexible -
Inner arrays don't have to be the same length

Jagged vs Multidimensional



Only Arrays

Simpler and more lightweight
– only one array



Any Collection

More flexible -
Inner arrays don't have to
gth

```
List< List<int> >
```

```
Dictionary<string, string[] >
```

Module Summary

- **Multidimensional Arrays**
  - More than one index
  - Great for grid-based data
- **Non-zero-based arrays**
  - Rarely encountered
- **Jagged Arrays**
  - Arrays of arrays
  - Concept applies to all collection types

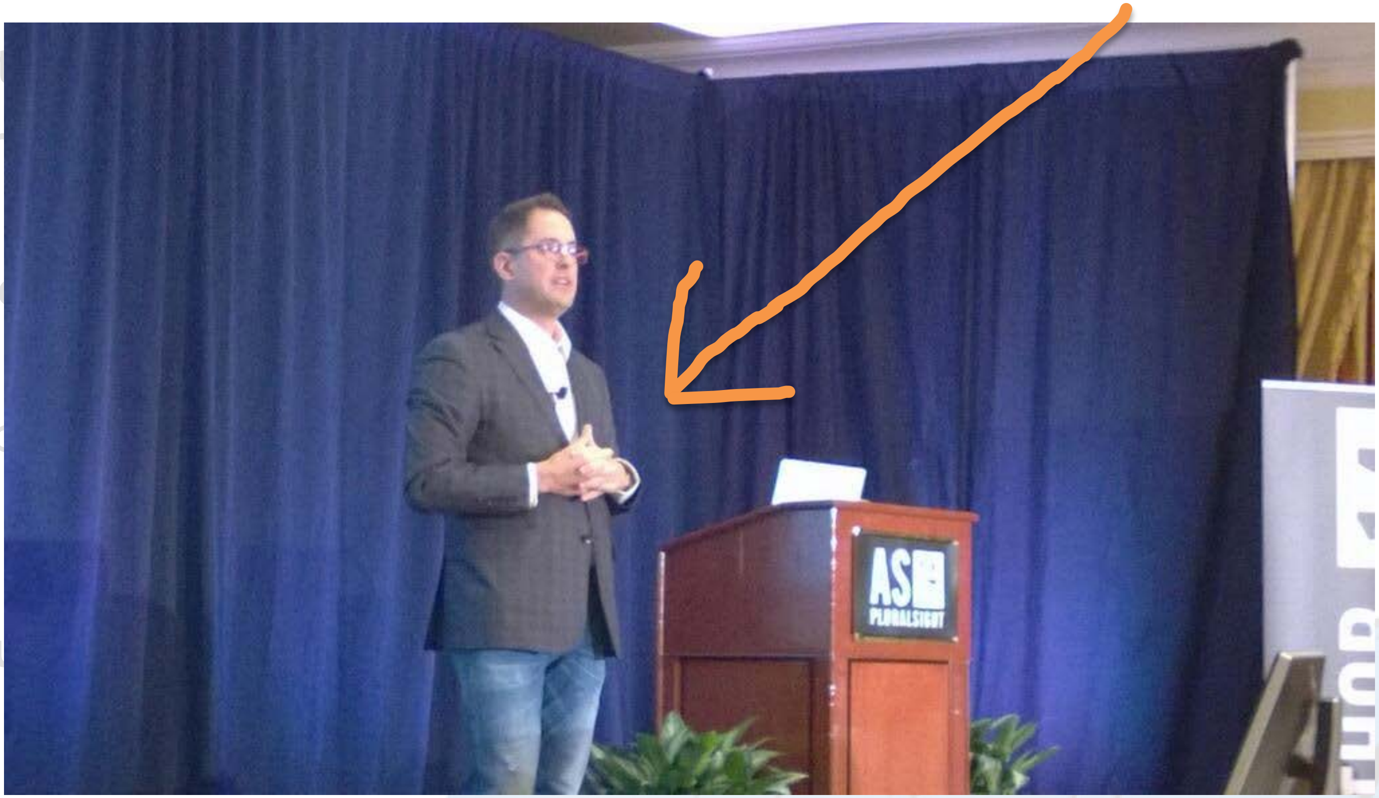
```
string[] daysOfWeek = {
                            "Monday",
                            "Tuesday",
                            "Wednesday",
                            "Thursday",
                            "Friday",
                            "Saturday",
                            "Sunday"
                        };
```

Course Summary

- Lists, Dictionaries and Sets
- Arrays implemented in CLR
  - Basis of most other collections
- Interfaces
  - `IEnumerator<T>`, `ICollection<T>`, etc.
- Types
  - `List<T>`, `Dictionary<TKey, TValue>`, etc.
  - Customizing some collections
- Enumerators

Course Summary

- List
- Arr
- Int
- Typ
- Enu



Aaron

Thanks to Pluralsight author Mike Erickson (author of *Introduction to UML*) for the picture,

C# Collections

Simon Robinson
http://TechieSimon.com
@TechieSimon