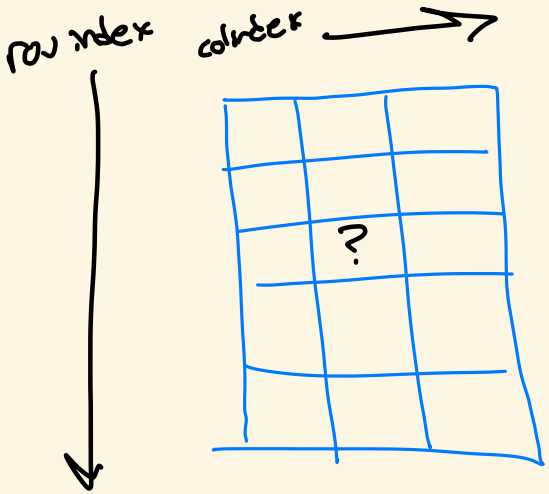
TOSTRING() METHODS

- What happens when you call?

```
System.out.println(myObject);
```

- Q. How does it know what to print?
- A. It calls the object's `toString()` method



Accessing values

1D: arr[2]: 3rd val

2D: arr[rowInd][colIndex]

arr[2][1]

MULTIDIMENSIONAL ARRAYS

- An array of arrays
- Ex: `int[][] arr = new int[5][3]`

rows cols rows cols

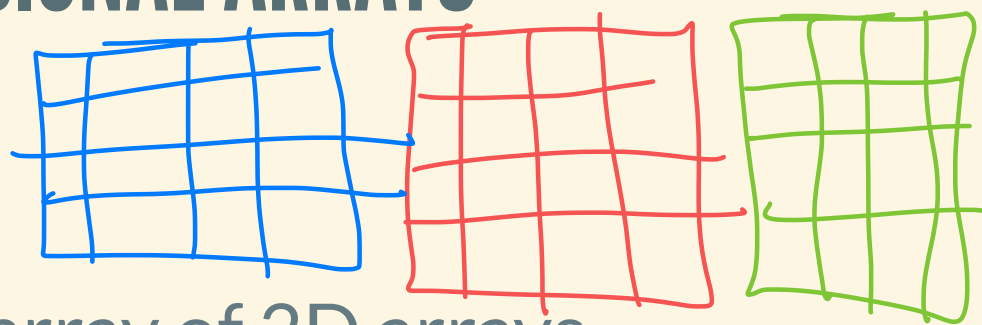
int[][][] arr3D = new int[5][3][8];

MULTIDIMENSIONAL ARRAYS

int, boolean, float

- Like 1D arrays can hold primitives, objects, etc.
- Purposes:
  - Representing grids (ex: game boards)
  - Representing tables/matrices
    - Physical simulations
    - ML/DM (representing samples and features)
    - images

MULTIDIMENSIONAL ARRAYS



- Not limited to 2D
- Can have 3D, 4D, etc.
- Ex: 3D array is just an array of 2D arrays
- Uses:
  - movie = sequence of images
  - color image can be thought of as N x M x 3 array

JAGGED ARRAYS

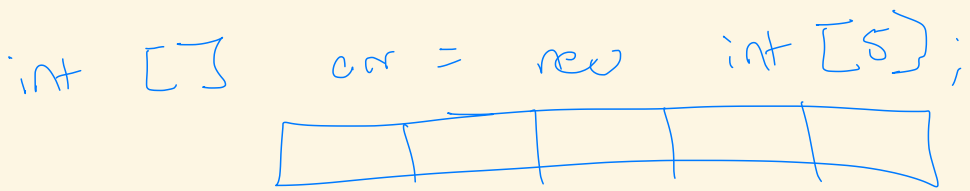
- `int[][] arr = new int[5][3];`
  - array of 5 arrays of size 3
- could create N-D array differently

```
int[][] arr = new int[2][];
```

- need to then create each internal array separately

ARRAYLISTS

- arrays are fixed size
- often may not know exactly how many (or max) items need to be stored
- ArrayList: basically a resizable array
- Can hold built-in or user defined objects
- **Cannot** store primitive types -> need to use capital version instead (aka `Integer` instead of `int`)



ARRAYLISTS CREATION

```
import java.util.ArrayList;
ArrayList<Integer> myvals = new ArrayList<Integer>();
```

name of ArrayList

- initially empty

ARRAYLIST METHODS

add (8)
add (0, 7)

- `add(Object o)`: add to end of list
- `add(int index, Object o)`: add to specified spot
- `get(int index)`: return element at specified spot
- `indexOf(Object o)`: index of 1st occurrence
- `size()`: number of elements in list
- `remove(int index)`: remove element at index
- `remove(Object o)`: remove 1st occurrence of specified object

remove (1)

SORTING ARRAYLISTS

- ArrayList is a type of "Collection"
- `Collections.sort(myvals);`
- there are other useful methods in `Collections`
- apply to things other than just ArrayLists