COS Crash Course: Week 3

arrays

name game

Yacoub Kahkajian *he/him*

Graduating in 2026

A.B. Computer Science

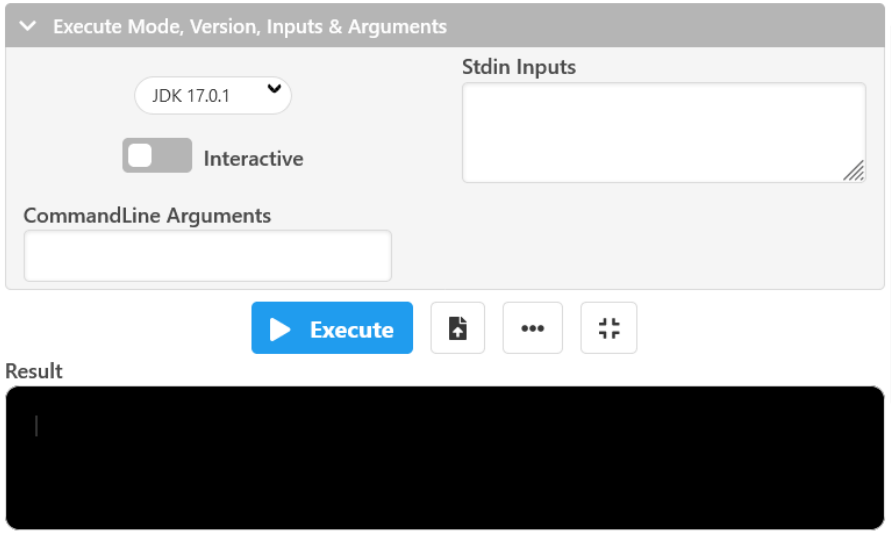
yacoub.xyz

jdoodle.com/online-java-compiler

Feel free to play around with it while I yap.

```java
public class MyClass {
    public static void main(String args[]) {
      int x=10;
      int y=25;
      int z=x+y;

      System.out.println("Sum of x+y = " + z);
    }
}
```



lecture

new dimensions

methods

2d arrays

into the 1st dimension

- Previously, when we wanted to store a value, we could only attribute it to one variable at a time.

```
int num = 2763;
String i = "ok";
```

- Arrays are examples of data structures: a single object that can contain a large set of data.
- Arrays store this data as an ordered, or indexed list of values. It's like going from a single point to a line!

```
int[] nums = {2, 7, 6, 3};
```

creating an array

- Java has two ways of creating a new array:
  - Create an empty array that can hold n values.

```
int[] foobar = new int[n];
```

  - Create an array with pre-defined values.

```
char[] alpha = {'a', 'b', 'c'};
```

- In Java, once you create a new array, you **CANNOT CHANGE ITS SIZE!** You can change existing values within it, but you CANNOT add values without copying the entire array to a new value.

computer memory

- Arrays play nicely with a computer's memory (RAM), since computer memory is kinda like a large array.
- Just like how an array is a sequence of indexed values, computer memory is a sequence of indexed locations.
- Taking a value from a specific index in an array is efficient, since the computer grabs it straight from its memory.

[2,7,6,3] →  →

2	7	6	3
a[0]	a[1]	a[2]	a[3]

array initialization

In your code editor, try initializing 4 new empty arrays of different data types with whatever length you'd like.

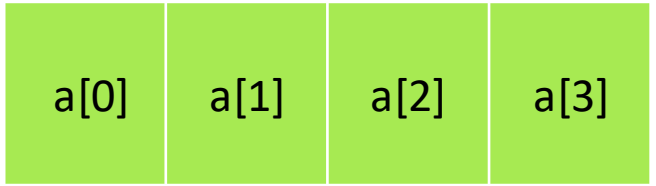
- Can you figure out how to print out the first element of each of these arrays? What is the default value of each empty array?
- How change you change this first element and reprint the new value?

boolean

int

double

String



lecture

new dimensions

methods

2d arrays

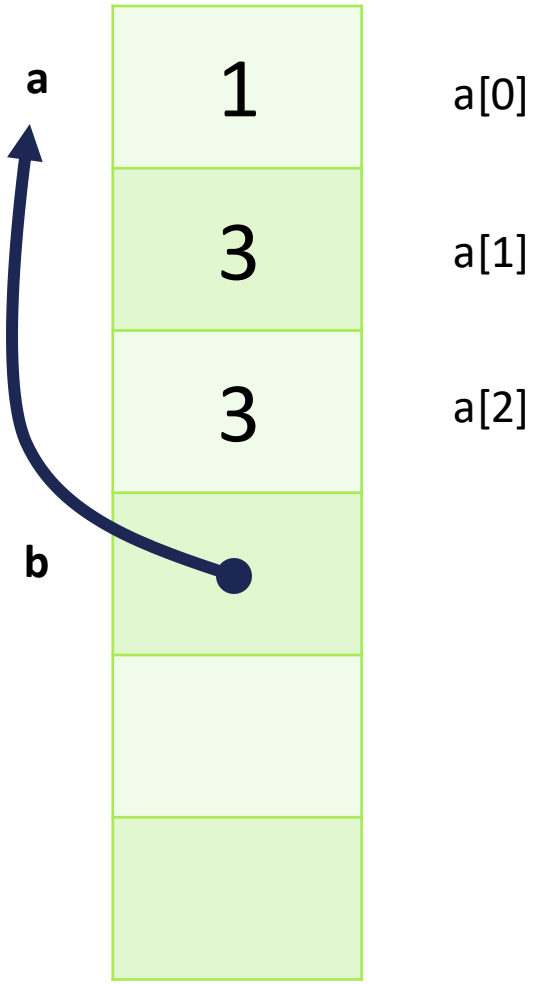
copying an array

- What if we wanted to make a copy of an existing array? Do we just do the same thing we do for single literal values?

```
int a = 5;
int b = a;
```

- No! Setting a new variable equal to a preexisting array only make that new variable *refer* to that array.
- It does NOT copy the array, so any changes made to the original array will be reflected when referring to the new variable.

```
int[] a = {1, 2, 3};
int[] b = a;
b[1]++;
System.out.print(b[1]);
System.out.print(a[1]);
```



copying an array (for real this time)

- To (actually) copy an array, you iterate through the array you want to copy and put them in a new, empty array.
- Copying an array isn't instant for this reason. It takes time proportional to the length of the array.

```
double[] b = new double[a.length];
for (int i = 0; i < a.length; i++)
    b[i] = a[i];
```

for each loops

- With a regular for loop, we can access each value in the array with nums[i] since indexes are stored in sequential order.

```
for (int i = 0; i < nums.length; i++)
```

- Or we can use a special for each loop to automatically copy the next value in a new variable for each iteration and refer to it with a custom name, num.

```
for (int num : nums)
```

```
int[] a = new int[Integer.parseInt(args[0])];
int[] b = new int[a.length];
b = a;
for (int i = 0; i < b.length; i++)
      b[i] = i;
for (int i = 0; i < a.length; i++)
      a[i] = 0;
for (num : a)
      System.out.print(num + " ");
System.out.println();
for (num : b)
      System.out.print(num + " ");
```

what does this code print? trace it!

lecture

new dimensions

methods

2d arrays

into the 2nd dimension

- We've seen how arrays can take our code to the first dimension. But what if we wanted to traverse the second and have a whole grid of values?
- 2d arrays store an array of arrays, which we can represent as a grid with x and y coordinates.

```
int[][] grid;

grid = new int[2][3];
```

a[0][0] 0	a[0][1] 0	a[0][2] 0
a[1][0] 0	a[1][1] 0	a[1][2] 0

double indexes

```
grid = new int[2][3];
```

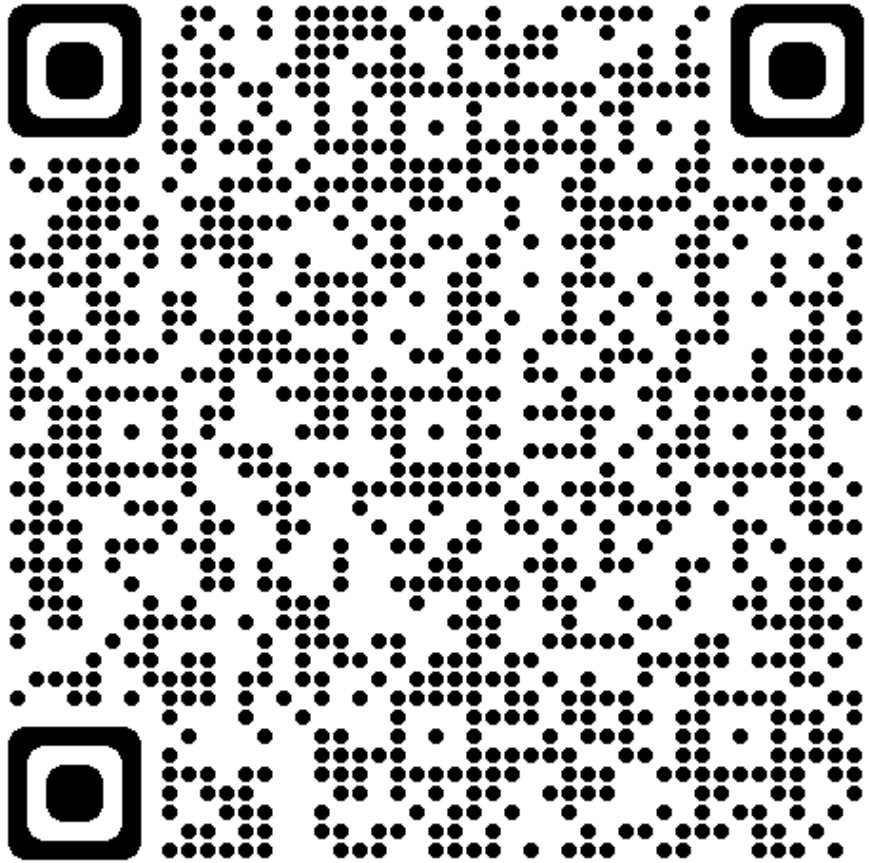
- 2d arrays are doubly indexed. You refer to rows by making that the first value in brackets, then columns by making that the second value.
- The code above creates a new 2d int array, named grid, that is 2 rows long and 3 columns wide.
- Double indexes still starts at zero. If we wanted to refer to the value at the bottom right corner of this array, we would type grid[1][2].

a[0][0] 0	**a[0][1]** 0	**a[0][2]** 0
a[1][0] 0	**a[1][1]** 0	**a[1][2]** 0

how can we print out every value in a 2d array?

hint: you'll need a nested for loop.

your input HALPS



exercises

two sum

Given an array of ints and a target number, find the indexes of two numbers such that the values stored within them add up to the target number.

stonks

Given an array of integer stock prices, where `prices[i]` represents the price of a stock on the ith day, find the maximum profit you can make within the date range if you could only buy stock one day and sell it during another.